



# GAMES 003 科研素养课

## 第一周：课程内容概览 & 建立领域视野

---



彭思达



高俊



彭崧猷



王倩倩

# 本课程的动机：针对低年级学生

## 低年级学生的常见情况

缺乏研究策略

面临科研困难

遭遇情绪低落 😞

## 学习我们的课程内容

如何选题与设计方法

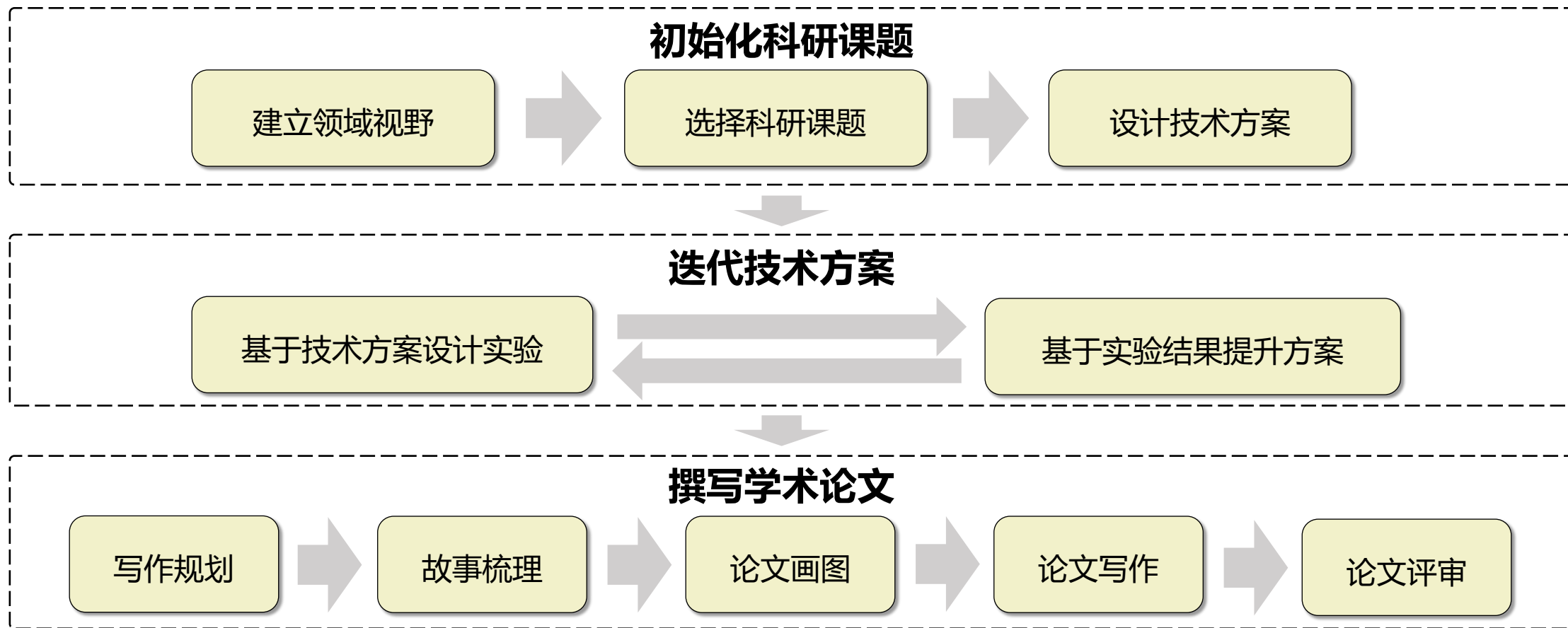
如何实验与提升方法

如何写作与宣传工作

知道如何在科研过程中自我提升科研能力，  
成为一个快乐的研究人员 😊

# 课程安排

- 传授科研流程，解释每一步的具体操作，配合案例加以说明。



# 课程安排

- 传授科研流程，解释每一步的具体操作，配合案例加以说明。
- **分享科学研究中的软技能，包括如何做报告和一些有用的研究习惯。**

## 掌握科研软技能

学术报告技巧

日常科研习惯

# 课程目标

对课程的预期

能教什么

阐明Research project的步骤和过程：

- 选题与Idea设计的流程、方法论；
- 进行实验的流程、方法论；
- 写论文的步骤流程。

解释规则，分享经验。

不能教什么

- 想Idea的能力（创新能力、技术能力）；
- 进行实验的能力（编程能力、分析能力、技术经验、沟通能力）；
- 论文写作能力（表达能力、英语水平、艺术水平、时间规划能力）。

这些能力只能通过实践经验来培养。

# 怎么最好地从这门课中受益

从本课程中学习科研流程与每个步骤的方法论



根据自己的经验思考并总结方法论

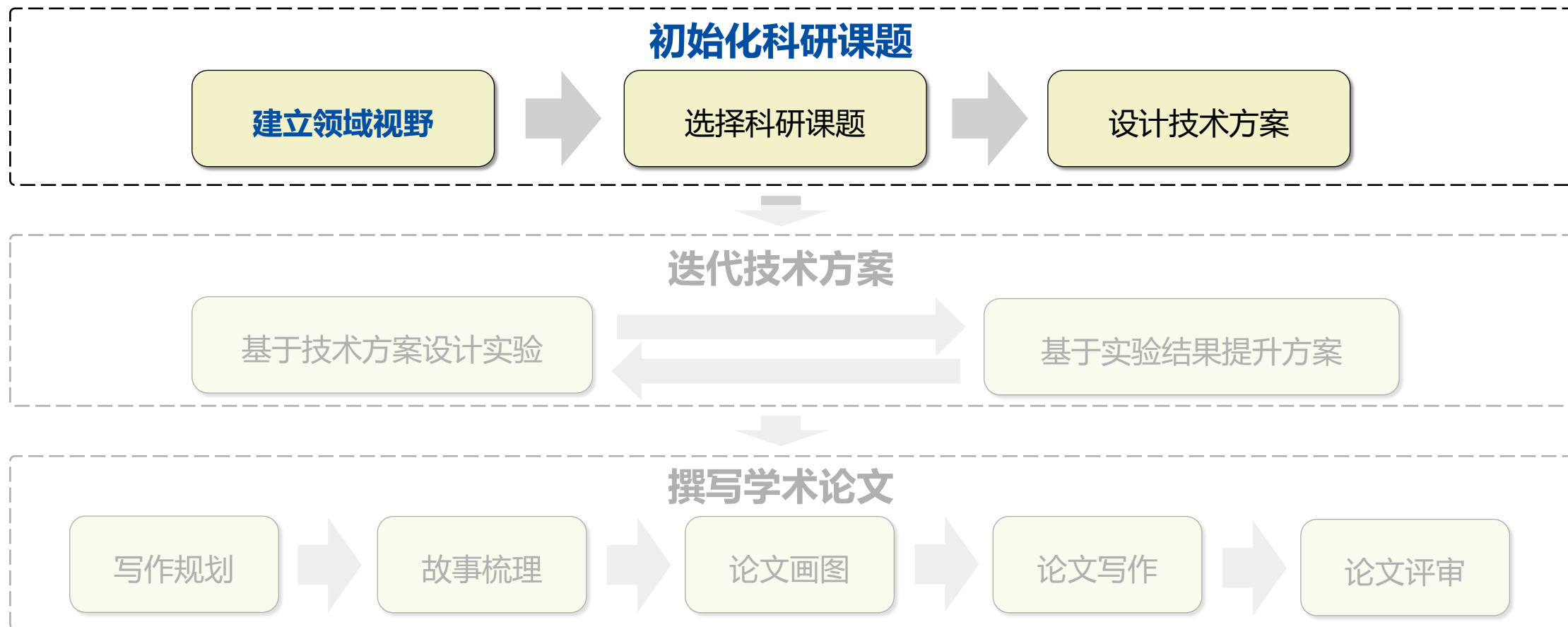


通过**实践**熟悉研究方法



**重要**：总结**你自己的**研究方法

# 第一部分：科研课题初始化



# 什么是领域的视野

“领域”的定义：特定的科研方向，如3D reconstruction、Novel view synthesis。

## 什么是领域的视野

### 对“技术演变”的视野

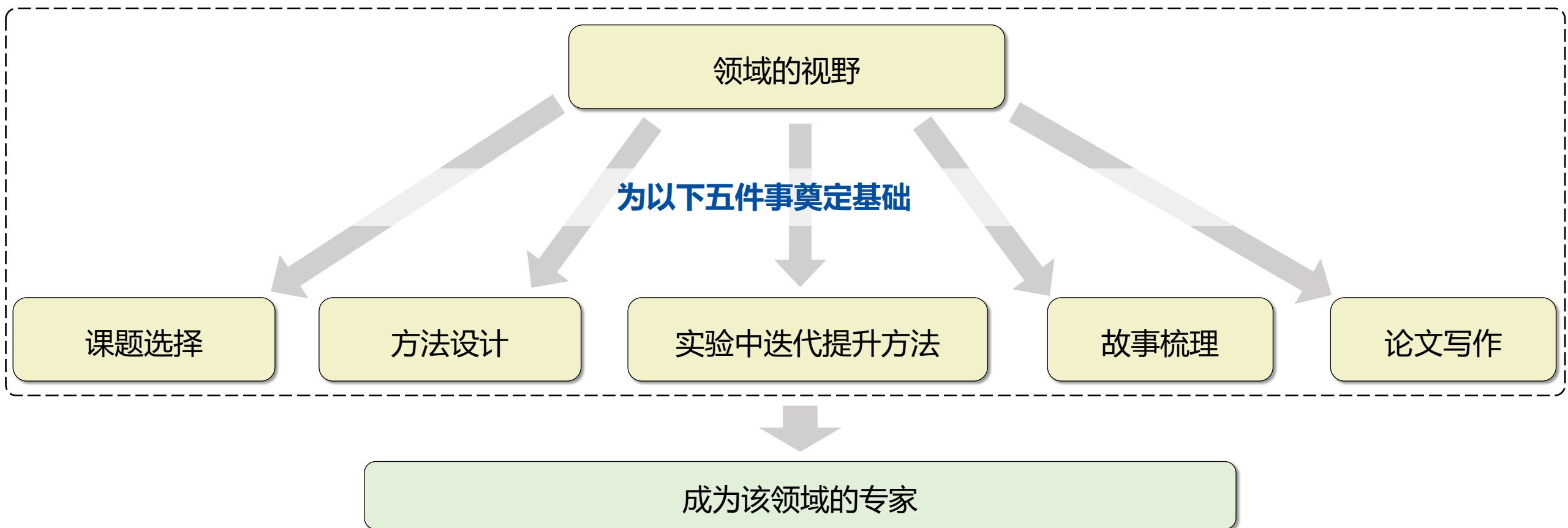
- 有哪些milestone papers？
- 领域技术如何随着时间的推移而演变？

### 对“重要问题”的视野

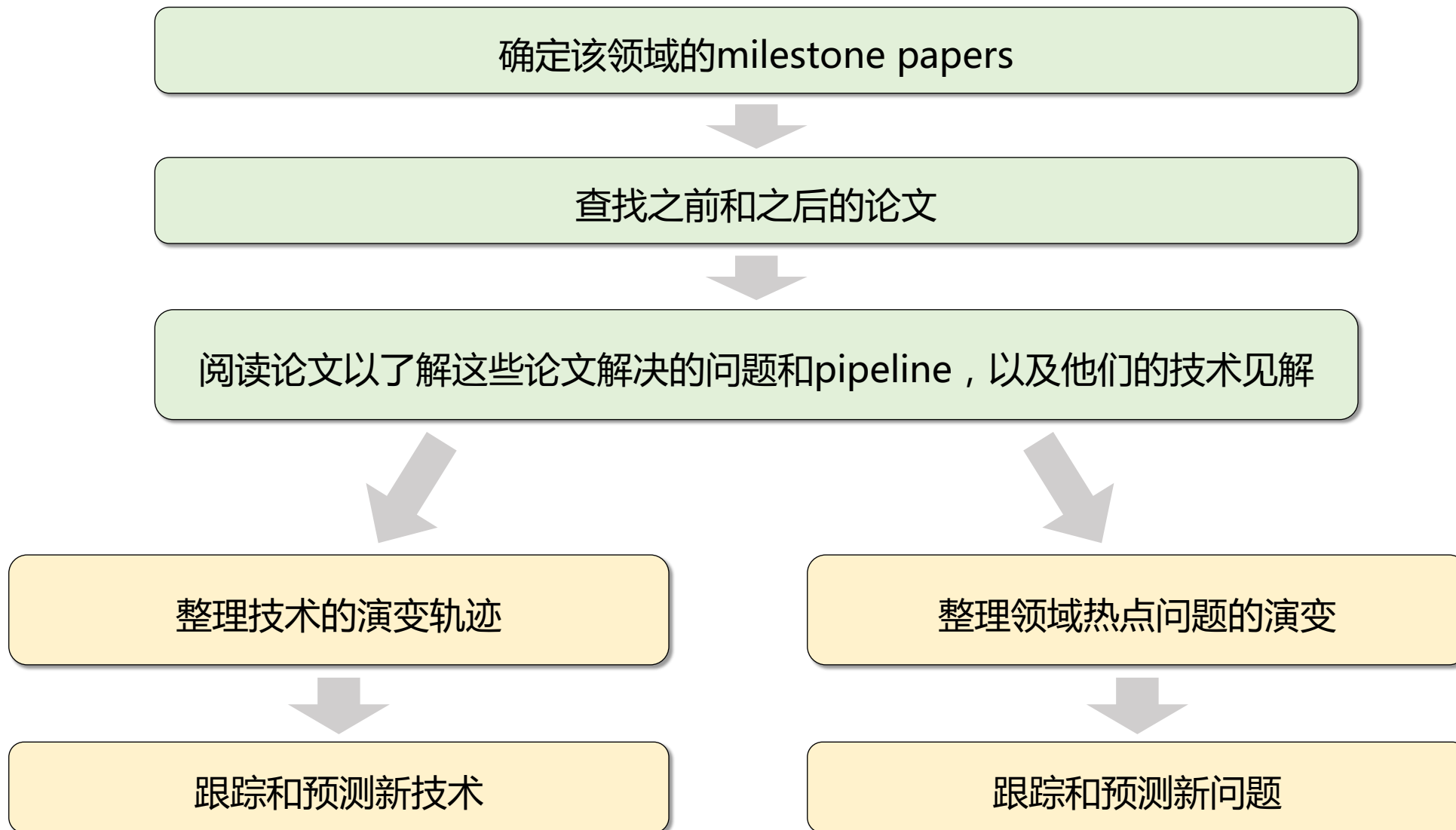
- 这个领域的终极目标是什么？
- 该领域已经达到了什么水平？
- 还有哪些重要的问题仍未被解决？
- 现阶段的热点话题是什么？



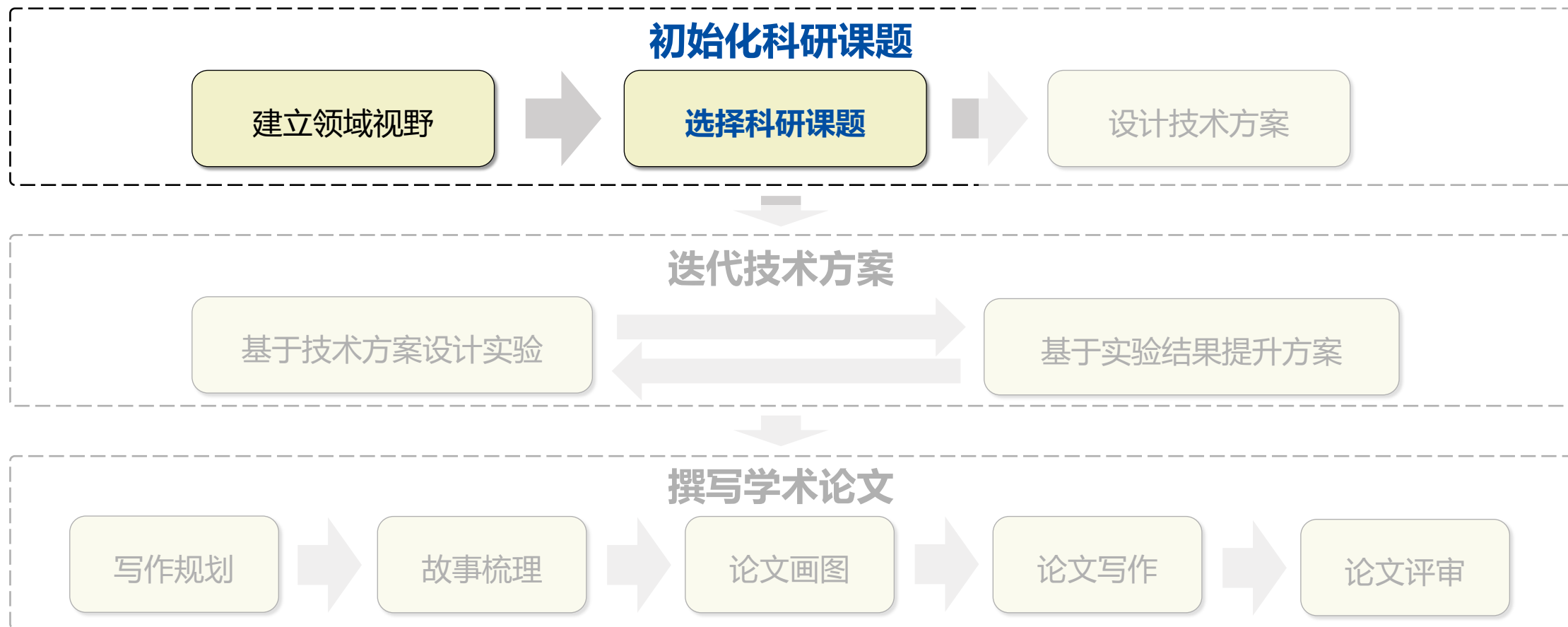
# 为什么我们需要建立视野



# 如何建立领域的视野



# 第一部分：科研课题初始化



# 课题选择

我们需要做什么

```
graph TD; A[我们需要做什么] --> B[确定Project setting (输入和输出)]; A --> C[确定想解决的failure case 及其背后的技术挑战];
```

确定Project setting  
(输入和输出)

确定想解决的failure case  
及其背后的技术挑战

# 为什么课题选择非常重要

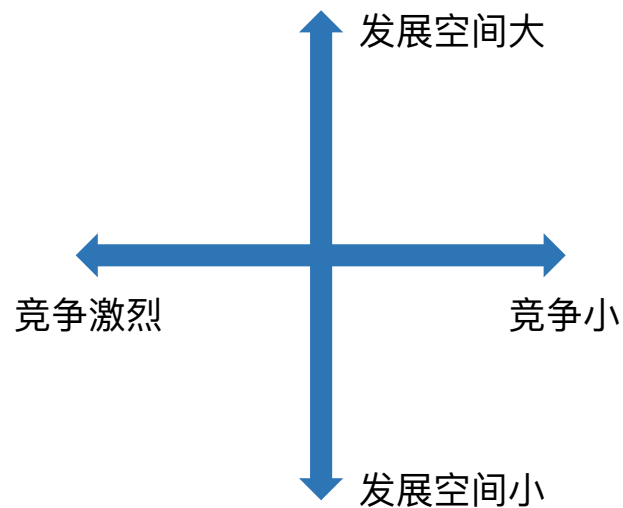
- 一个好的课题将使科研变得容易很多。

## 什么是好的课题

发展空间大、竞争程度小

任务难度与个人能力相匹配

该课题的影响大



如果任务太难，可能会导致无法完成。  
如果任务太简单，可能没法锻炼自己的技术能力。

考虑该项目可以产生的学术影响程度。

# 如何找到好的课题

## 寻找好课题的四个步骤

### 建立领域的视野

- 整理milestone papers/技术范式。
- 了解它们是如何随着时间演变的。

### 列出领域内的课题

- 列出该领域的重要问题。
- 判断哪个问题是当前时间点应该去解决的。

### 判断哪些课题是好课题

- 评估课题的竞争程度和发展空间。
- 考虑任务的难度与个人能力相匹配。

### 在选题上Aim High

- 考虑任务的影响力。
- 技术风险性大的课题往往有更高的收益。

**Aim high: 努力完成一个达到你能力极限的科研课题。**

# 不同阶段的学生应该如何选择科研课题

Junior students, Senior students



Junior students, Senior students



Junior students

Senior students

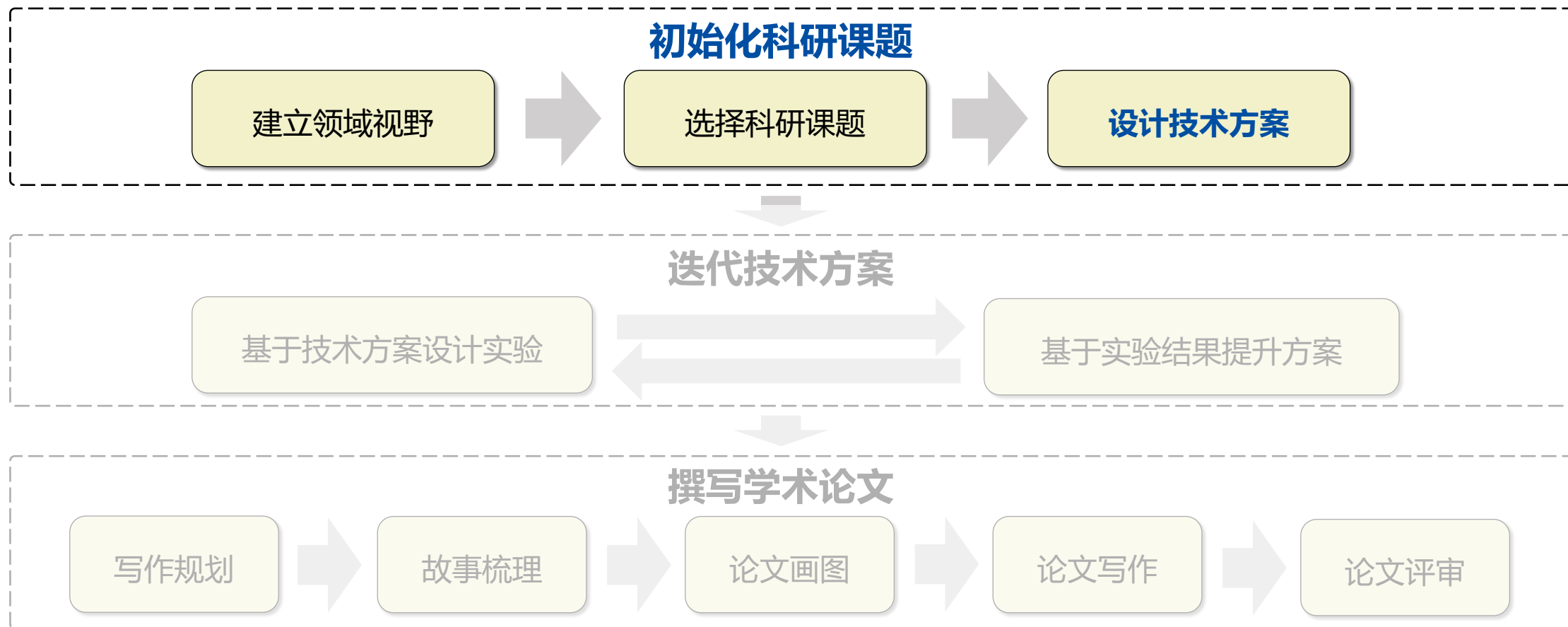


Junior students, Senior students

Senior students



# 第一部分：科研课题初始化





# 方法设计

## 什么是方法设计

- 在选定课题后，我们需要设计一种新颖的pipeline来达到SOTA performance。

## 为什么要有方法设计的系统性思路

- 一个清楚的方法设计思路能更有效地解决问题并达到SOTA。
- 让我们提出的方法更有动机，同时使得论文的故事更吸引人。
- 保证了方法的技术创新性，以免审稿人说它没有novelty。
- 同时，该系统性思路可以有效提升我们的技术insights。

# 如何设计方法

## 分析原因

- 遵从第一性原理，分析为什么当前的SOTA效果不够好。

## 设计方法

- 根据分析的原因，设计一种能达到SOTA的方法。

## 判断合理性

- 判断提出的技术方法是否合理

## 改进方法

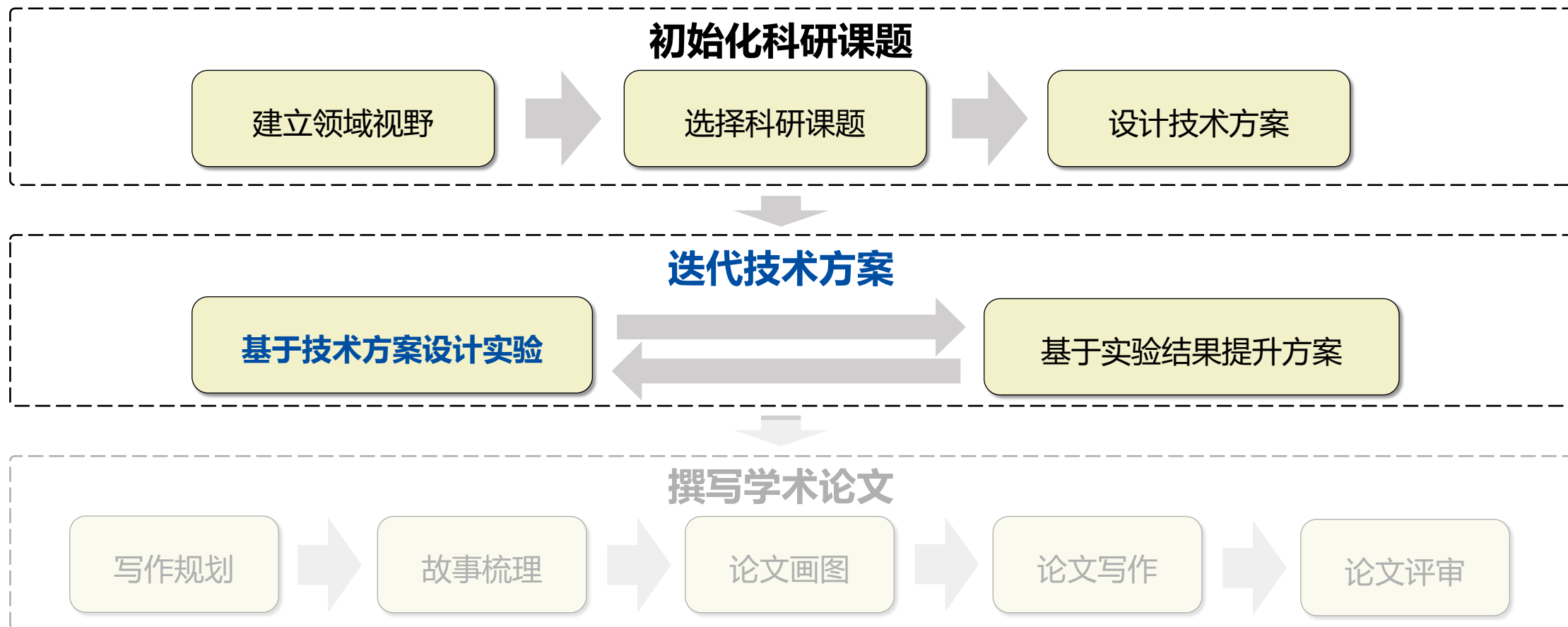
- 通过读论文、讨论和做实验改进技术方法

# 如何设计方法

## 三个要点

- 不只是一要Work，还要保证论文方法的**技术创新性**。
- 争取在Pipeline层面设计解法。
- 要判断论文方法是否有足够技术贡献，是否有新的知识、新的认识、新的观点。

## 第二部分：实验中迭代提升方法



# 实验设计

## 实验设计是什么

- 规划要做哪些实验，实现并提升Technical idea，达到SOTA指标。

## 实验设计的两个好处

- 让研究变得更简单：想清楚实验的目的，明确科学研究的方法。
- 让实验更简单：降低实验难度，提高实验效率。

# 如何设计实验

- 核心原则：减少实验中包含的探索点的数量。

如何减少exploration points的数量

分解pipeline

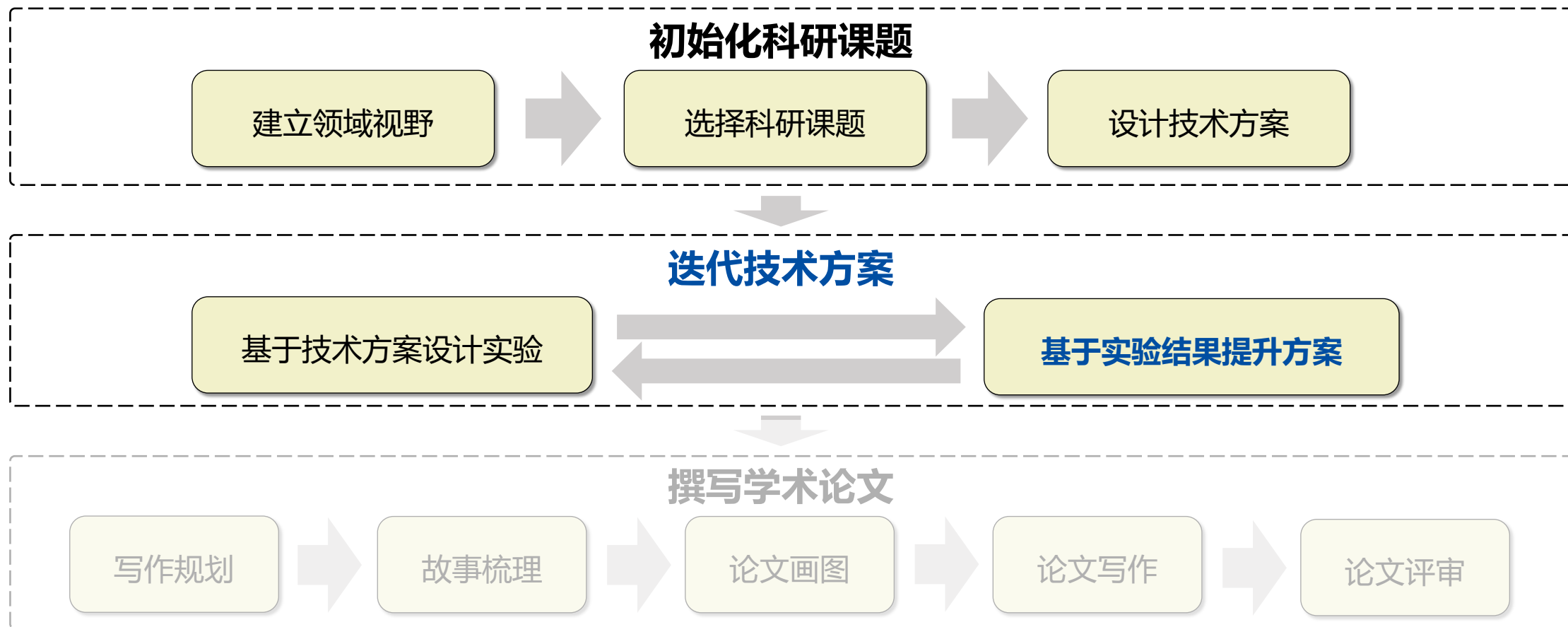
将idea分解为不同的组件，从可控的idea开始，不断添加探索性和创新性的框架/模块。

分解实验设置

从一个简单的setting开始探索，然后逐渐增加难度，之后进入到真正的setting。

我们还应该考虑exploration points的重要性，进行实验优先级排序

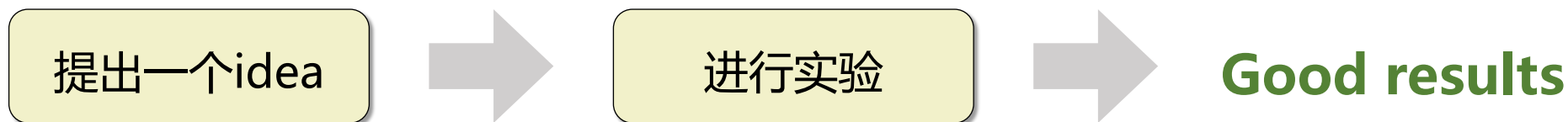
## 第二部分：实验中迭代提升方法



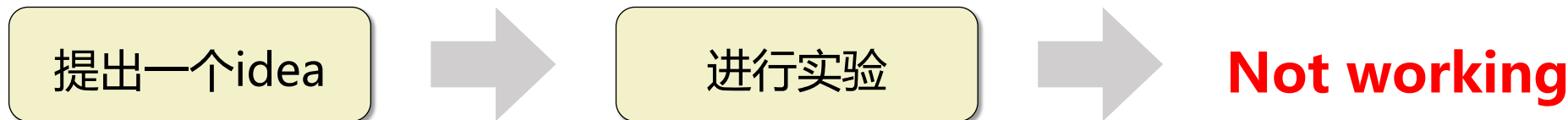
# 为什么需要改进方法

- 因为初始提出的方法通常会不work。

理想情况：



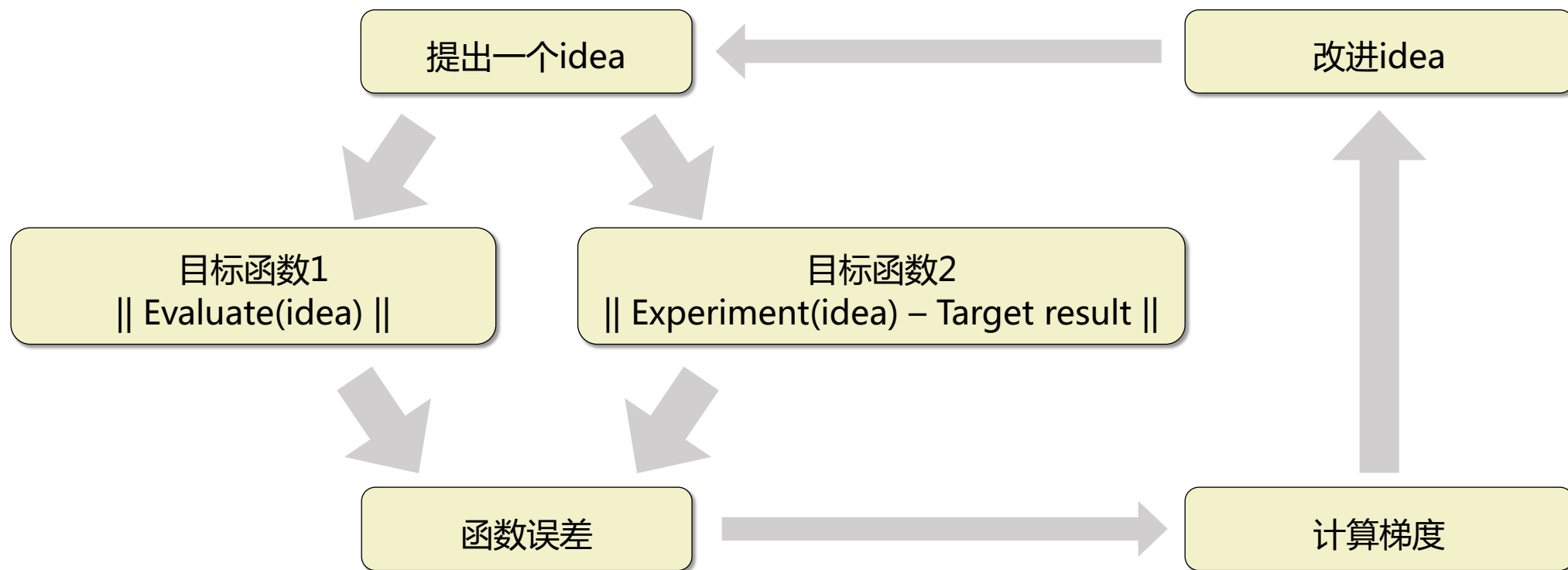
实际情况：



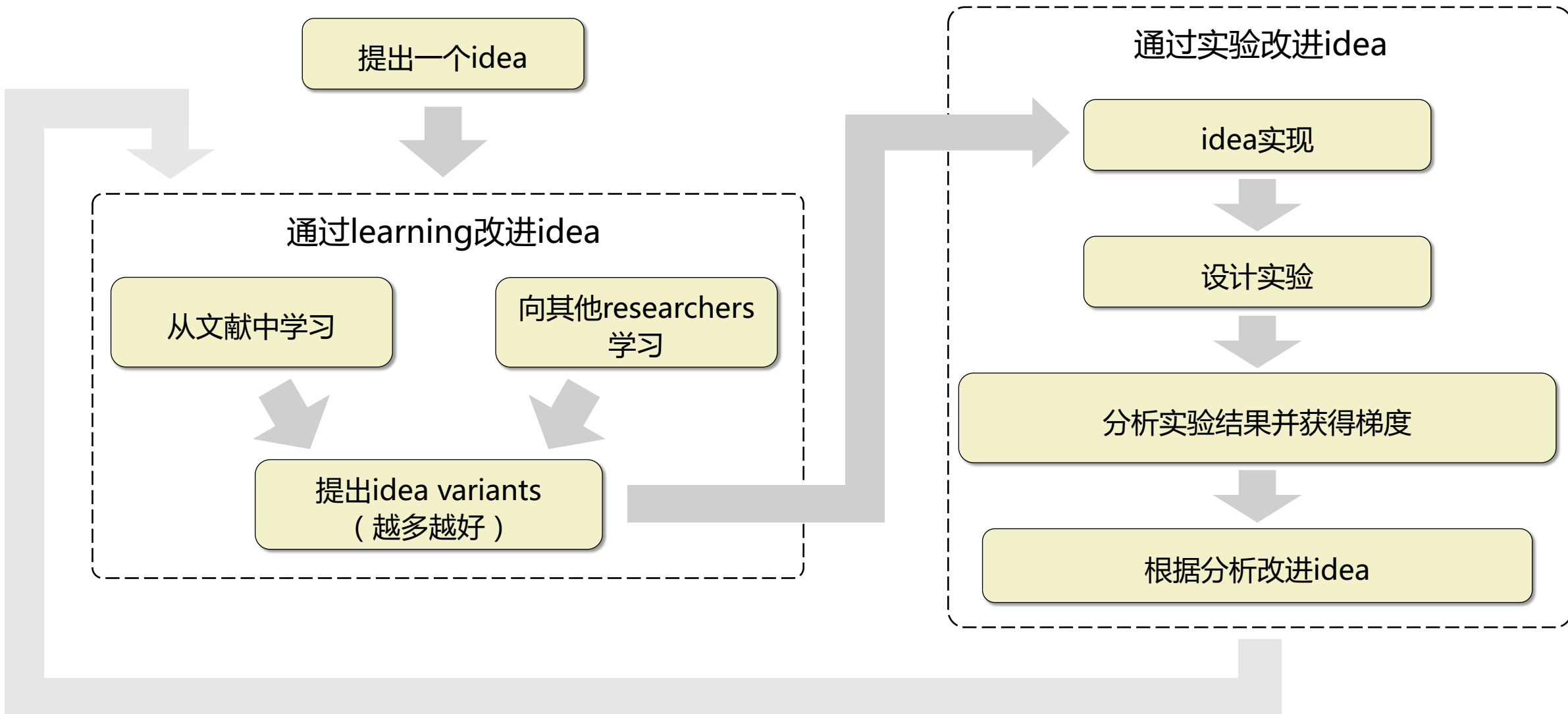


# 如何改进方法

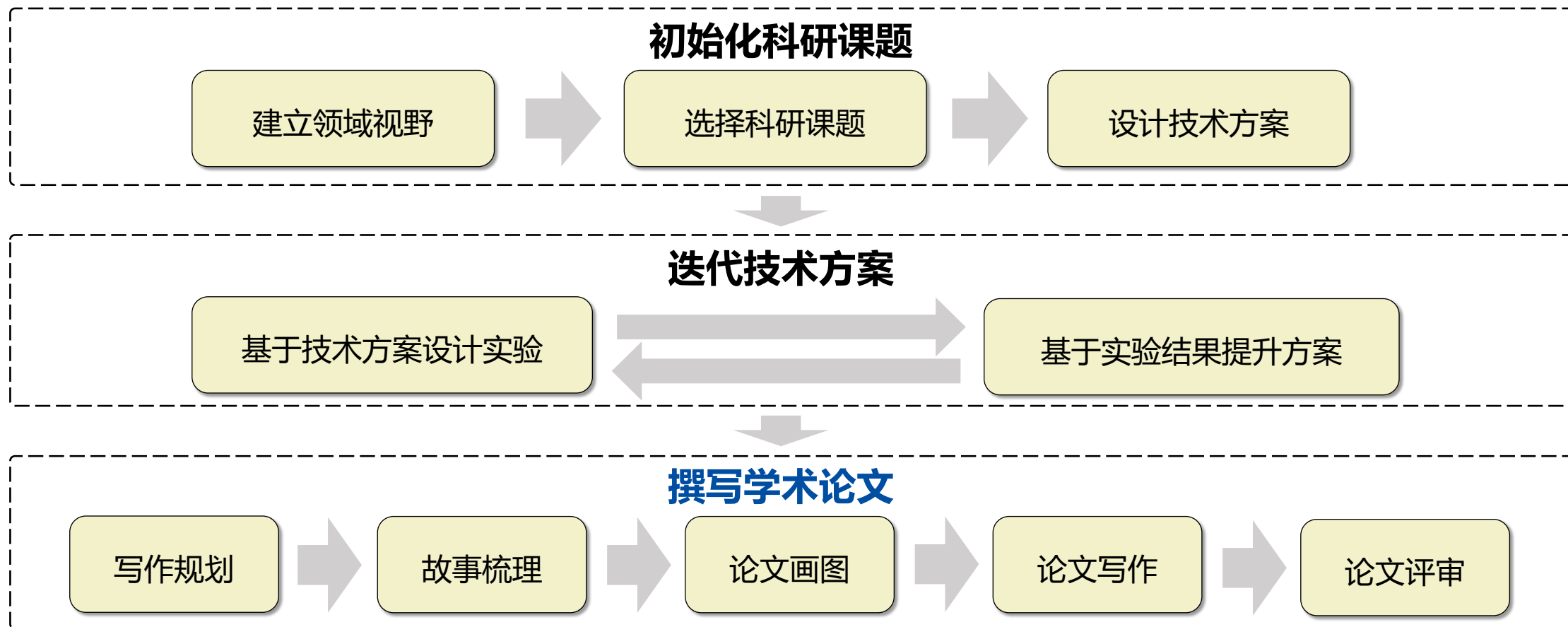
- 可以把改进方法的过程当作SGD优化过程。



# 如何改进方法



# 第三部分：论文写作



# 写作规划

Step 1 画一个清楚的pipeline figure的草图（理清楚方法的流程步骤），梳理论文story，写一版introduction的初稿。

Step 2 列出要做的comparison experiments和ablation studies，并开始做起来。

Step 3 写method，同时做实验。

Step 4 改introduction和method，同时做实验。

Step 5 实验做差不多以后，写experiment。

Step 6 写related work。

Step 7 Review论文。改论文的introduction、method和experiment。

Step 8 写abstract，取论文名字。

Step 9 反复review论文，改论文。

# 为什么我们需要写作规划

## 为什么规划很重要

- 大幅**提高投稿率**!!!
- 让自己做论文更轻松，**避免最后几天熬夜**。

# 怎么梳理论文的故事

## 首先，整理论文方法流程

给论文画一个pipeline figure的草图，帮助自己直观地看清论文方法：

- 按照“输入→模块1→中间输出1→模块2→.....→模块N→输出”的风格，用文字描述pipeline。（**我们需要清楚地描述 pipeline**）。
- 选择流程图的布局并绘制简单的流程图。

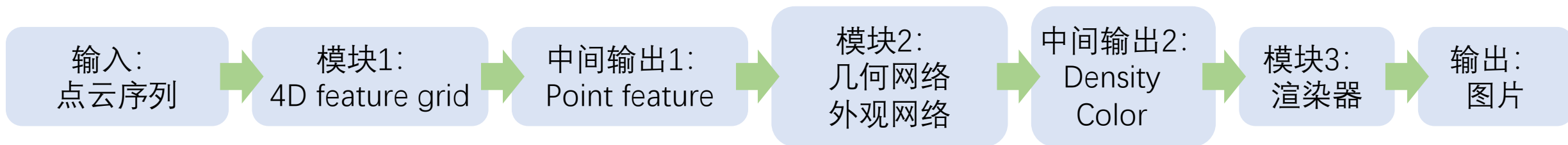


# 案例学习：4K4D

- 用文字描述pipeline
  - 输入：点云序列
  - 模块1：4D feature grid
  - 中间输出1：每个点的feature
  - 模块2：几何网络、外观网络
  - 中间输出2：每个点的density、color
  - 模块3：点云渲染器
  - 输出：图片

# 案例学习：4K4D

- 选择流程图的布局并绘制简单的流程图





# 怎么梳理论文的故事

## 第二，按序回答以下问题

1. 我们的pipeline有哪些贡献（提出新任务、确定新的技术挑战、提出新的技术贡献）？
2. 我们的贡献有哪些好处？它们解决了哪些技术挑战？
3. 我们的论文带来了哪些新的insights？
4. 我们如何通过介绍以往的方法引入我们关注的技术挑战和新的insights？

## 第三，概述论文故事

1. 介绍论文的任务。
2. 通过讨论以前的方法来引入我们解决的技术挑战。
3. 为了应对这一技术挑战，我们提出 xx 贡献。
4. 我们的贡献有哪些技术优势，有哪些新的insights？

# 论文画图



画好看的图的目的: **提高中稿率**

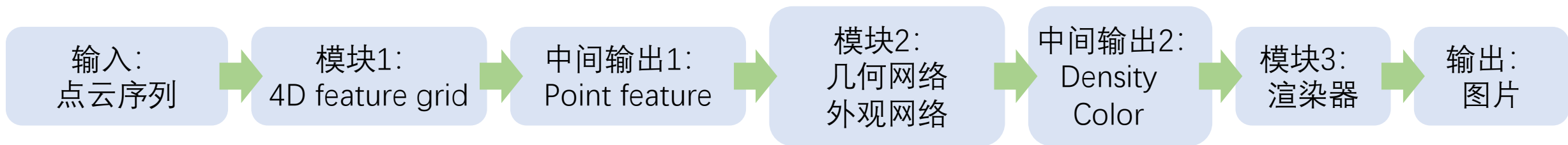
# 怎么论文画图

## 绘制pipeline图的步骤流程

- 按照“输入→模块1→中间输出1→模块2→.....→模块N→输出”的风格，用文字描述pipeline。（**我们需要清楚地描述 pipeline**）。
- 选择流程图的布局并绘制简单的流程图。
- 可视化“输入”、“模块”、“中间输出”和“输出”等元素。
- 给流程图涂上颜色。
- 优化流程图布局，保证简洁。避免过多的空白区域和大的空白区域。

# 案例学习：4K4D

- 选择流程图的布局并绘制简单的流程图



# 案例学习：4K4D

- 好看的线条、框图、元素可视化、配色非常重要

输入：  
点云序列

模块1：  
4D feature grid

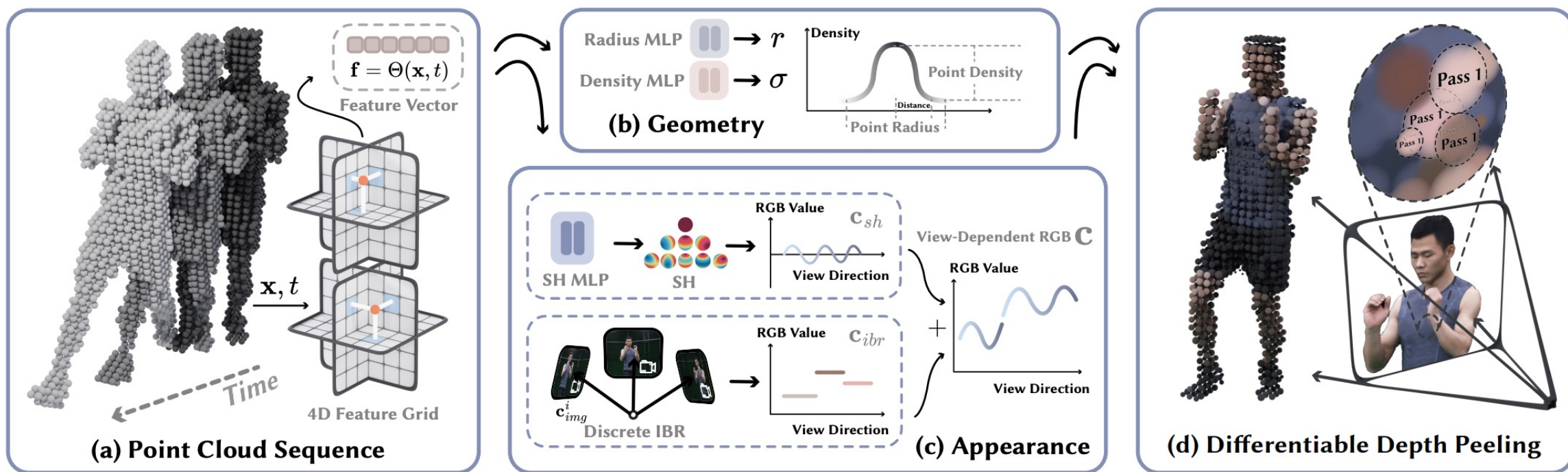
中间输出1：  
Point feature

模块2：  
几何网络  
外观网络

中间输出2：  
Density  
Color

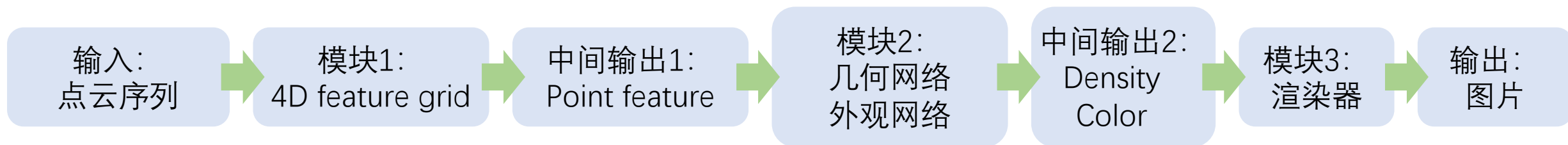
模块3：  
渲染器

输出：  
图片



# 案例学习：4K4D

- 好看的**线条**、**框图**、元素可视化、配色非常重要



# 案例学习：4K4D

- 好看的线条、框图、**元素可视化**、配色非常重要

输入：  
点云序列

模块1：  
4D feature grid

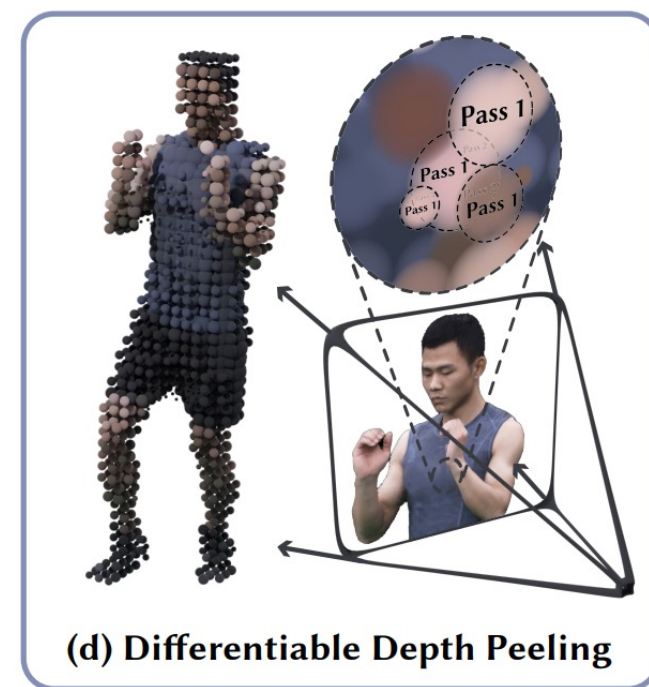
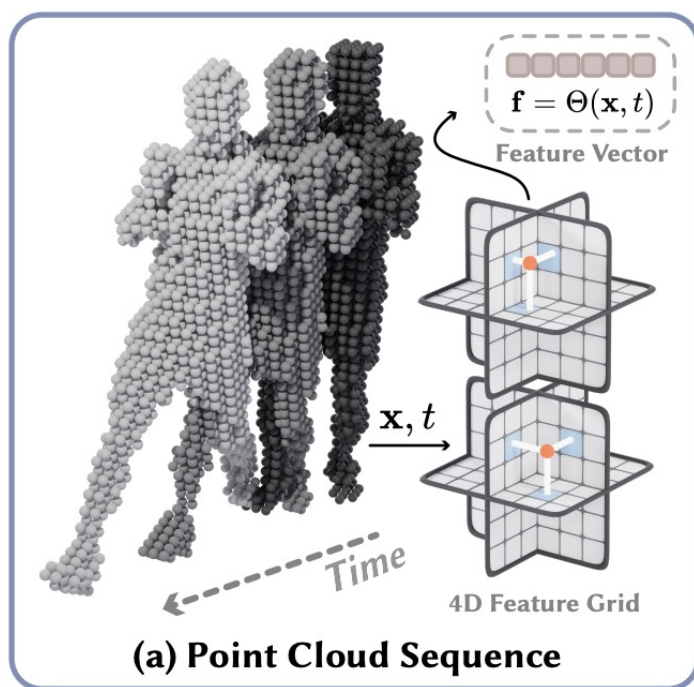
中间输出1：  
Point feature

模块2：  
几何网络  
外观网络

中间输出2：  
Density  
Color

模块3：  
渲染器

输出：  
图片





# 案例学习：4K4D

- 好看的线条、框图、元素可视化、**配色**非常重要

输入：  
点云序列

模块1：  
4D feature grid

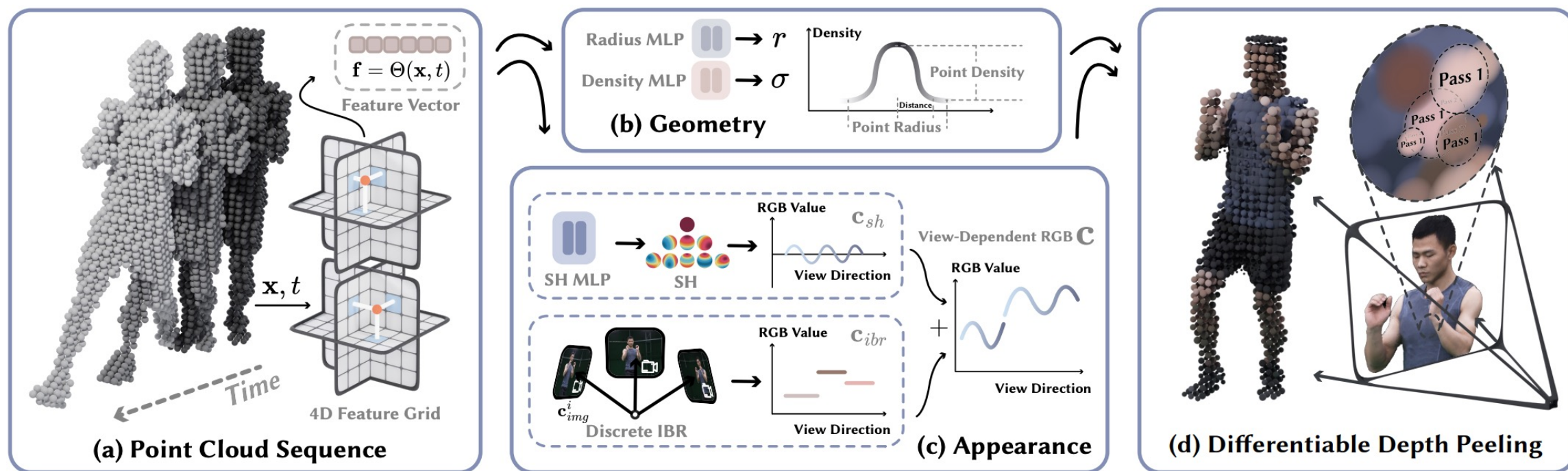
中间输出1：  
Point feature

模块2：  
几何网络  
外观网络

中间输出2：  
Density  
Color

模块3：  
渲染器

输出：  
图片





# 怎么写论文的方法

## 写method的步骤

- 画一个pipeline figure的草图。
- 梳理Method section，确定各个subsection将涵盖哪个方法模块。
- 分别写各个subsection。每个subsection包括三个部分：**模块的具体设计**、**模块的motivation**、**模块的技术优势**。
  - 首先写清楚模块的具体设计：给定xxx输入，第一步做xxx，第二步做xxx，第三步做xxx，最终得到xxx输出。
  - 然后模块的motivation、技术优势。

# 案例学习 : Neural Body

Module design:  
介绍module中的  
data structure

Technical  
advantages  
of this module

## 3.1. Structured latent codes

To control the spatial locations of latent codes with the human pose, we anchor these latent codes to a deformable human body model (SMPL) [38]. SMPL is a skinned vertex-based model, which is defined as a function of shape parameters, pose parameters, and a rigid transformation relative to the SMPL coordinate system. The function outputs a posed 3D mesh with 6890 vertices. Specifically, we define a set of latent codes  $\mathcal{Z} = \{z_1, z_2, \dots, z_{6890}\}$  on vertices of the SMPL model. For the frame  $t$ , SMPL parameters  $S_t$  are estimated from the multi-view images  $\{\mathcal{I}_t^c | c = 1, \dots, N_c\}$  using [26]. The spatial locations of the latent codes are then transformed based on the human pose  $S_t$  for the density and color regression. Figure 3 shows an example. The dimension of latent code  $z$  is set to 16 in our experiments.

Similar to the local implicit representations [25, 5, 18], the latent codes are used with a neural network to represent the local geometry and appearance of a human. Anchoring these codes to a deformable model enables us to represent a dynamic human. With the dynamic human representation, we establish a latent variable model that maps the same set of latent codes to the implicit fields of density and color at different frames, which naturally integrates observations.

# 案例学习 : Neural Body

Motivation  
of this module

Module design:  
通过介绍module  
forward process  
来介绍module design

## 3.2. Code diffusion

Figure 3(a) shows the process of code diffusion. The implicit fields assign the density and color to each point in the 3D space, which requires us to query the latent codes at continuous 3D locations. This can be achieved with the trilinear interpolation. However, since the structured latent codes are relatively sparse in the 3D space, directly interpolating the latent codes leads to zero vectors at most 3D points. To solve this problem, we diffuse the latent codes defined on the surface to nearby 3D space.

Inspired by [65, 56, 49], we choose the SparseConvNet [21] to efficiently process the structured latent codes, whose architecture is described in Table 1. Specifically, based on the SMPL parameters, we compute the 3D bounding box of the human and divide the box into small voxels with voxel size of  $5mm \times 5mm \times 5mm$ . The latent code of a non-empty voxel is the mean of latent codes of SMPL vertices inside this voxel. SparseConvNet utilizes 3D sparse convolutions to process the input volume and output latent code volumes with  $2\times, 4\times, 8\times, 16\times$  downsampled sizes. With the convolution and downsampling, the input codes are diffused to nearby space. Following [56], for any point in 3D space, we interpolate the latent codes from multi-scale code volumes of network layers 5, 9, 13, 17, and concatenate them into the final latent code. Since the code diffusion should not be affected by the human position and orientation in the world coordinate system, we transform the code locations to the SMPL coordinate system.

For any point  $\mathbf{x}$  in 3D space, we query its latent code from the latent code volume. Specifically, the point  $\mathbf{x}$  is first transformed to the SMPL coordinate system, which aligns the point and the latent code volume in 3D space. Then, the latent code is computed using the trilinear interpolation. For the SMPL parameters  $S_t$ , we denote the latent code at point  $\mathbf{x}$  as  $\psi(\mathbf{x}, \mathcal{Z}, S_t)$ . The code vector is passed into MLP networks to predict the density and color for point  $\mathbf{x}$ .

# 案例学习 : Neural Body

## 3.3. Density and color regression

Figure 3(b) overviews the regression of density and color for any point in 3D space. The density and color fields are represented by MLP networks. Details of network architectures are described in the supplementary material.

**Density model.** For the frame  $t$ , the volume density at point  $\mathbf{x}$  is predicted as a function of only the latent code  $\psi(\mathbf{x}, \mathcal{Z}, S_t)$ , which is defined as:

$$\sigma_t(\mathbf{x}) = M_\sigma(\psi(\mathbf{x}, \mathcal{Z}, S_t)), \quad (1)$$

where  $M_\sigma$  represents an MLP network with four layers.

**Color model.** Similar to [37, 44], we take both the latent code  $\psi(\mathbf{x}, \mathcal{Z}, S_t)$  and the viewing direction  $\mathbf{d}$  as input for the color regression. To model the location-dependent incident light, the color model also takes the spatial location  $\mathbf{x}$  as input. We observe that temporally-varying factors affect the human appearance, such as secondary lighting and self-shadowing. Inspired by the auto-decoder [48], we assign a latent embedding  $\ell_t$  for each video frame  $t$  to encode the temporally-varying factors.

Specifically, for the frame  $t$ , the color at  $\mathbf{x}$  is predicted as a function of the latent code  $\psi(\mathbf{x}, \mathcal{Z}, S_t)$ , the viewing direction  $\mathbf{d}$ , the spatial location  $\mathbf{x}$ , and the latent embedding  $\ell_t$ . Following [51, 44], we apply the positional encoding to both the viewing direction  $\mathbf{d}$  and the spatial location  $\mathbf{x}$ , which enables better learning of high frequency functions. The color model at frame  $t$  is defined as:

$$\mathbf{c}_t(\mathbf{x}) = M_c(\psi(\mathbf{x}, \mathcal{Z}, S_t), \gamma_d(\mathbf{d}), \gamma_x(\mathbf{x}), \ell_t), \quad (2)$$

where  $M_c$  represents an MLP network with two layers, and  $\gamma_d$  and  $\gamma_x$  are positional encoding functions for viewing direction and spatial location, respectively. We set the dimension of  $\ell_t$  to 128 in experiments.

Module design:  
通过介绍module  
forward process  
来介绍module design

Module design:  
介绍module中的  
data structure

Module design:  
通过介绍module  
forward process  
来介绍module design

# 论文自我评审

## 评审论文时我们要做什么

- 确定审稿人可能指出的潜在问题。

## 为什么要自我评审

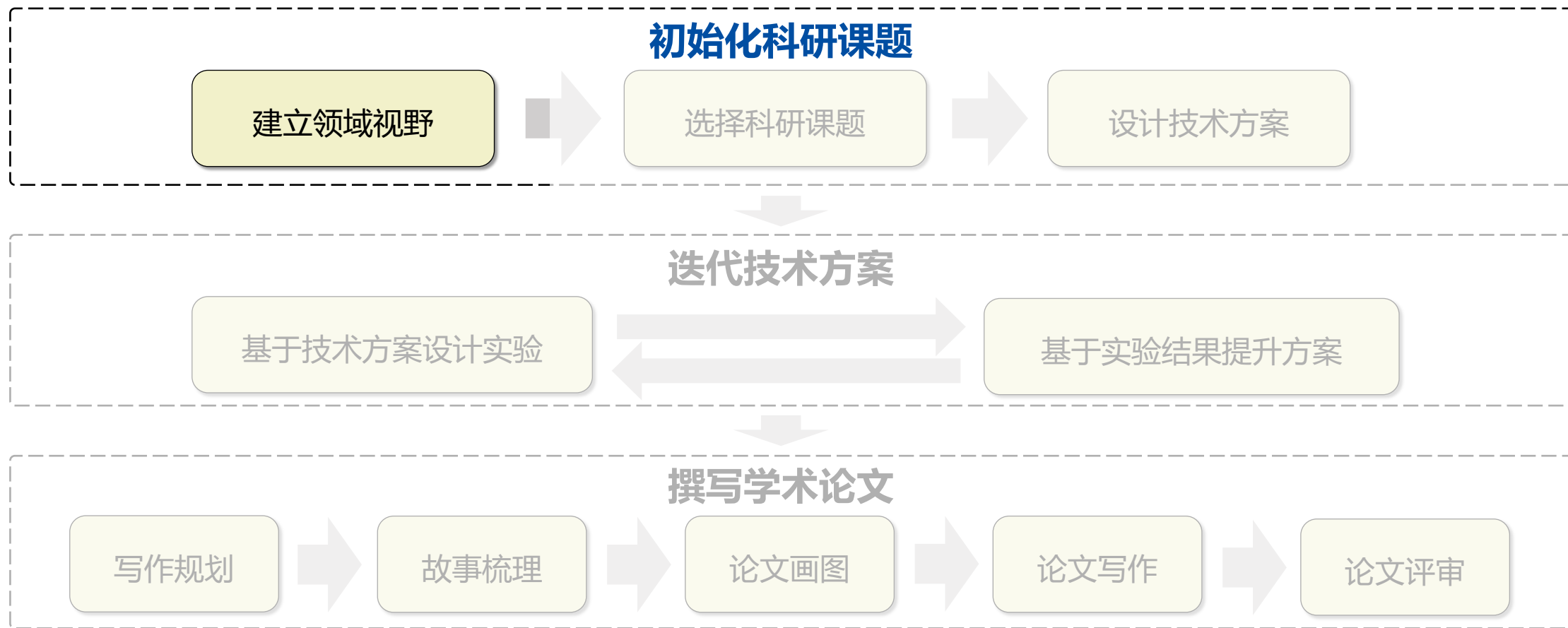
- 以提高论文的中稿率。



# 如何自我评审论文：A checklist

1. 技术贡献是否足够	1.1 想解决的failure cases很常见
	1.2 提出的技术已经被well-explored了，该技术带来的performance improvement是可预见的/well-known的
2. 论文写作是否清楚	2.1 论文Introduction是否清楚描述论文贡献
	2.2 论文pipeline figure是否清楚描述pipeline与技术贡献
	2.3 缺少技术细节，不可复现
	2.4 每个方法模块是否都写了motivation
3. 实验效果是否足够好	3.1 是否比之前方法好很多
	3.2 实验效果是否让人impressive
4. 实验测试是否充分	4.1 缺少ablation studies
	4.2 缺少重要的baselines、缺少重要的evaluation metric
	4.3 数据太简单，无法证明方法是否真的work
5. 方法设计是否合理	5.1 实验的setting不实际
	5.2 方法存在技术缺陷，看起来不合理
	5.3 方法模块的鲁棒性：是否需要在每个场景上调超参
	5.4 新的方法设计在带来benefit的同时，引入了更强的limitation，导致新方法的收益为负

# 建立领域视野的具体指导



# 什么是领域的视野

“领域”的定义：特定的科研方向，如3D reconstruction、Novel view synthesis。

## 什么是领域的视野

### 对“技术演变”的视野

- 有哪些milestone papers？
- 领域技术如何随着时间的推移而演变？

### 对“重要问题”的视野

- 这个领域的终极目标是什么？
- 该领域已经达到了什么水平？
- 还有哪些重要的问题仍未被解决？
- 现阶段的热点话题是什么？

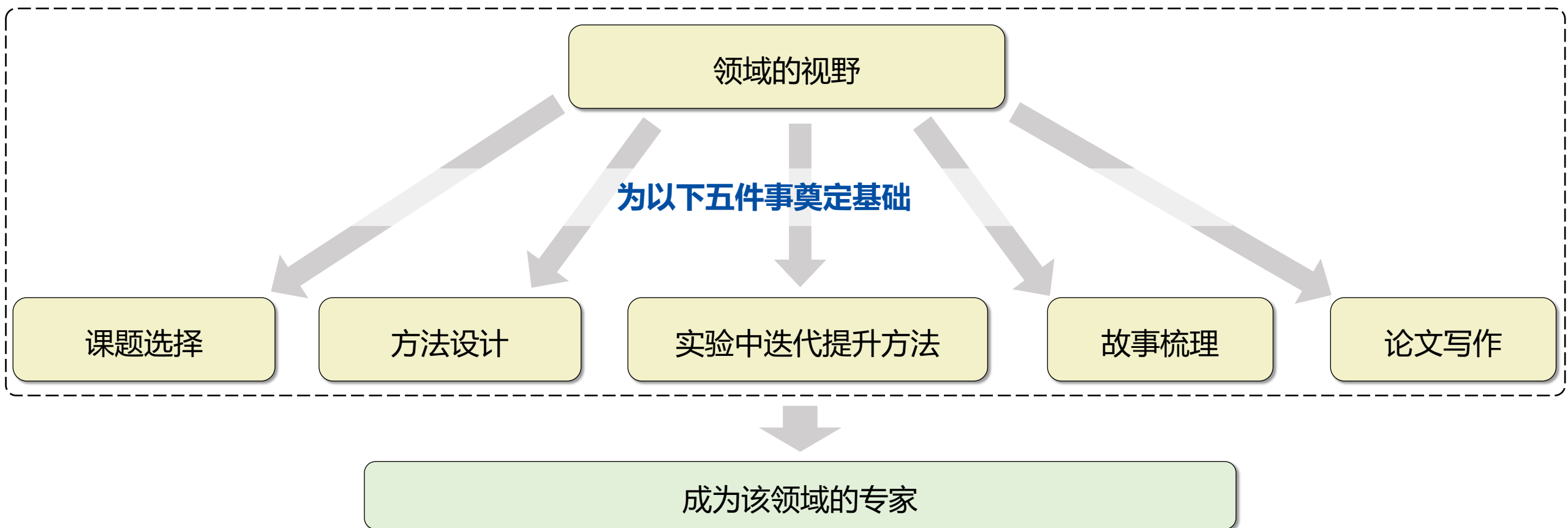




# 案例学习：多视角重建

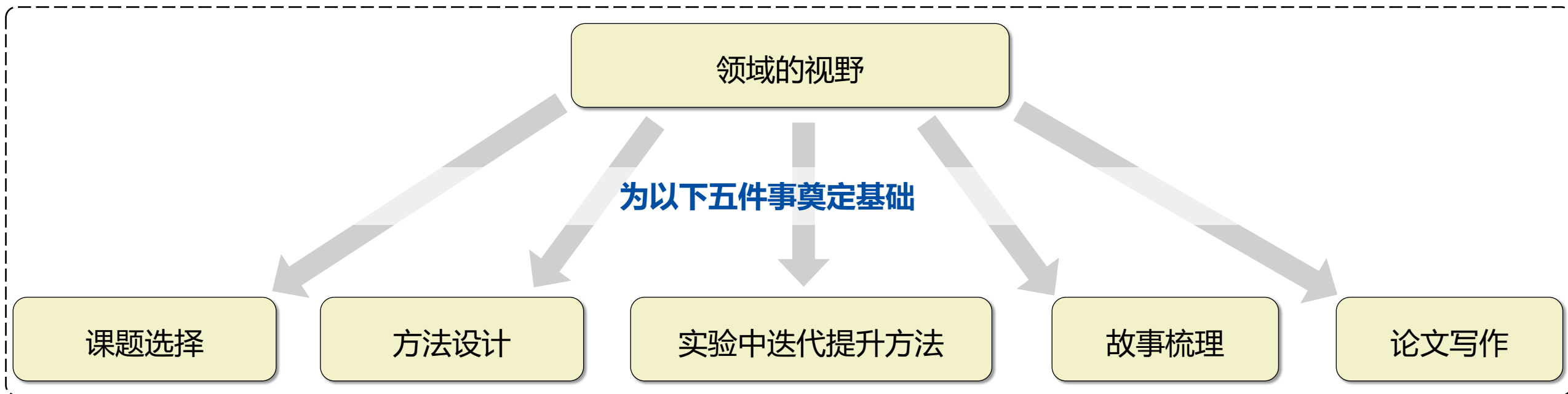
- 多视角重建的技术如何随着时间的推移而演变？
- 该方向还存在哪些重要的问题？

# 为什么我们需要建立视野

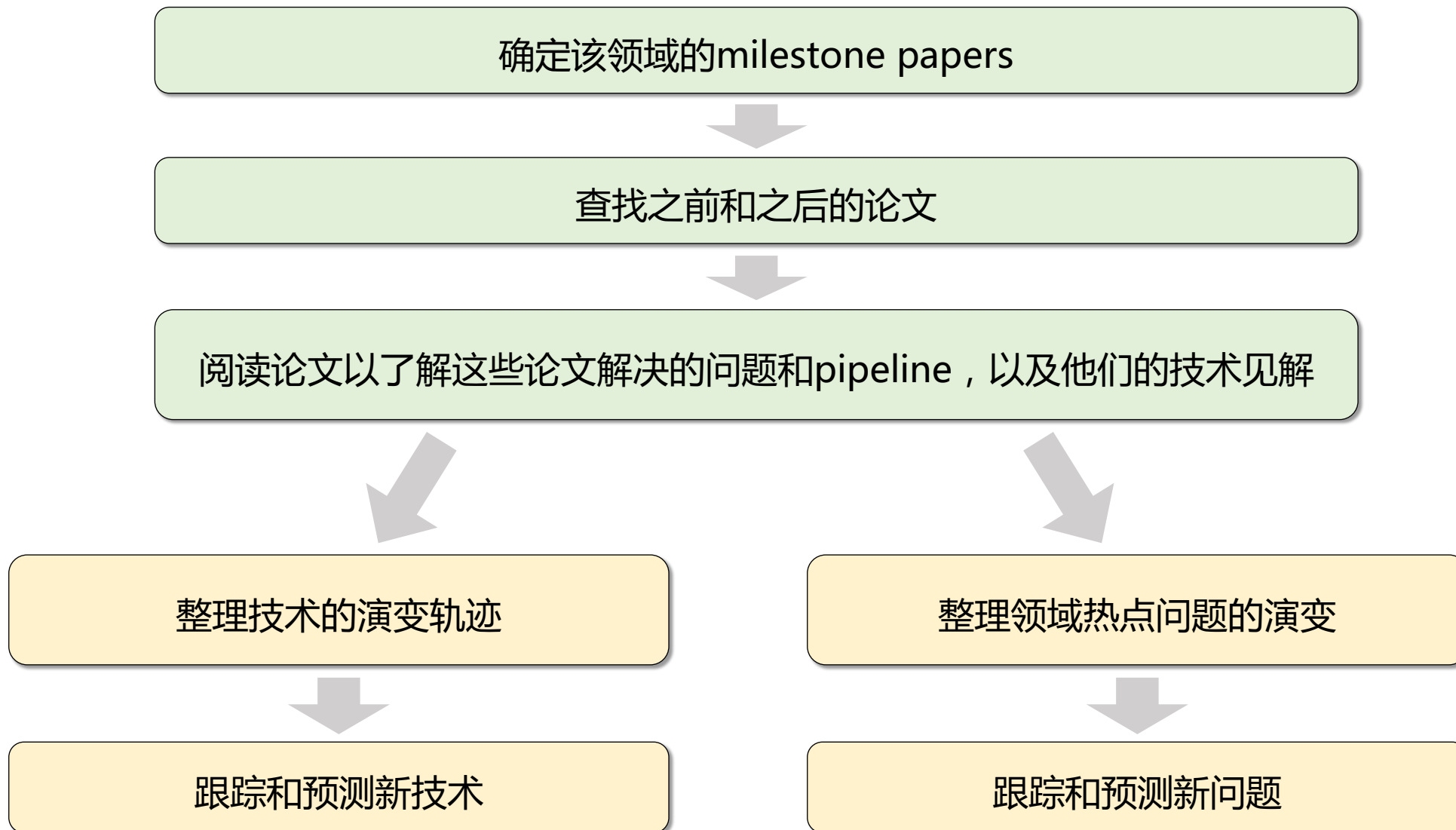


# 案例学习：多视角重建

- 多视角重建的技术如何随着时间的推移而演变？
  - 2023年：One-2-3-45
- 该方向还存在哪些重要的问题？
  - 2023年：稀疏视角重建



# 如何建立领域的视野



# 案例学习：多视角重建

- 2003—2016 , Colmap : Multi-view stereo
- 2018—2020 , MVSNet : 使用3D CNN在cost volume融合
- 2020—2023 , NeRF : 定义global representation , 使用differentiable rendering融合
- 2023—2024 , DUS3R : 将图像tokenize后 , 使用vit融合
- 2023—2024 , CAT3D : 使用Multi-view Diffusion Model融合

# 如何识别Milestone papers

## 两个技巧

- 查找该领域被引用次数最多的论文并按日期对它们进行排序。
- 寻求经验丰富的研究人员的建议。

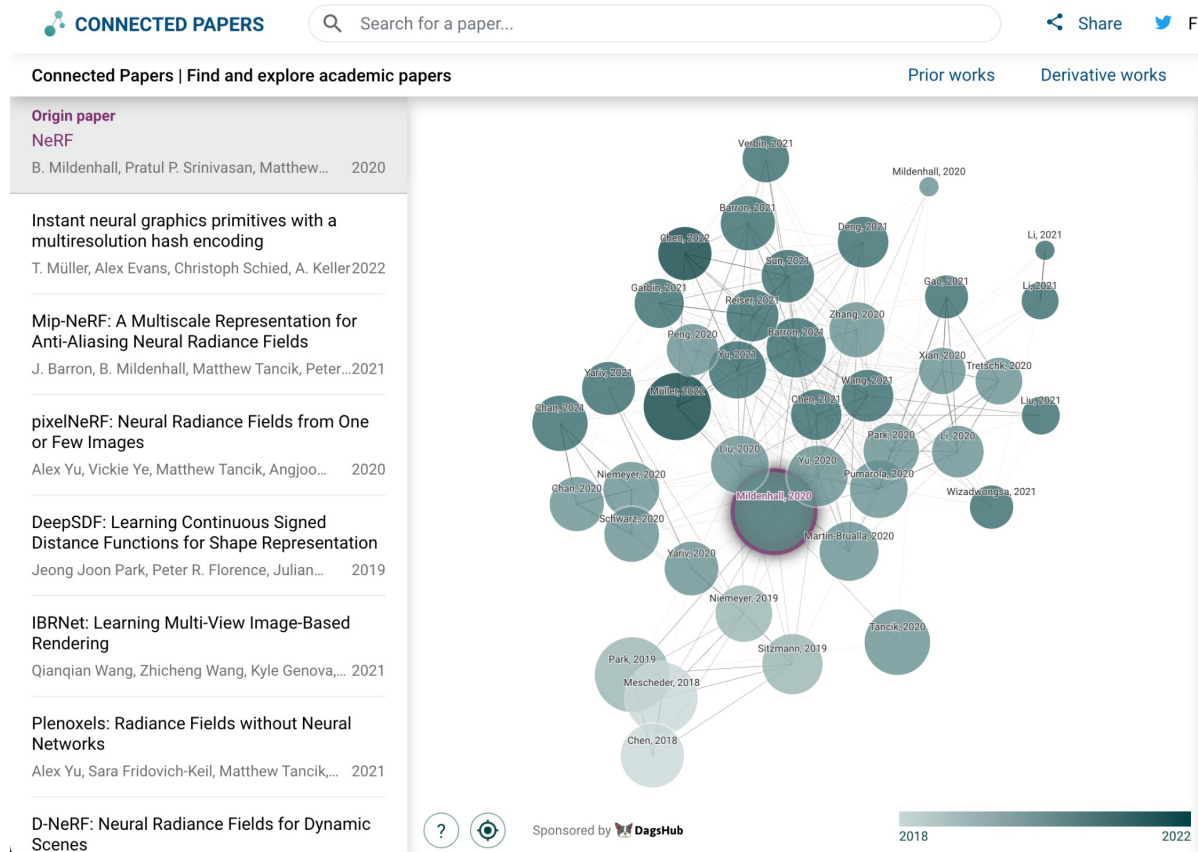
# 如何查找两个技术里程碑之间的论文

## 向前追溯

- 里程碑论文引用的论文
- 基础工作

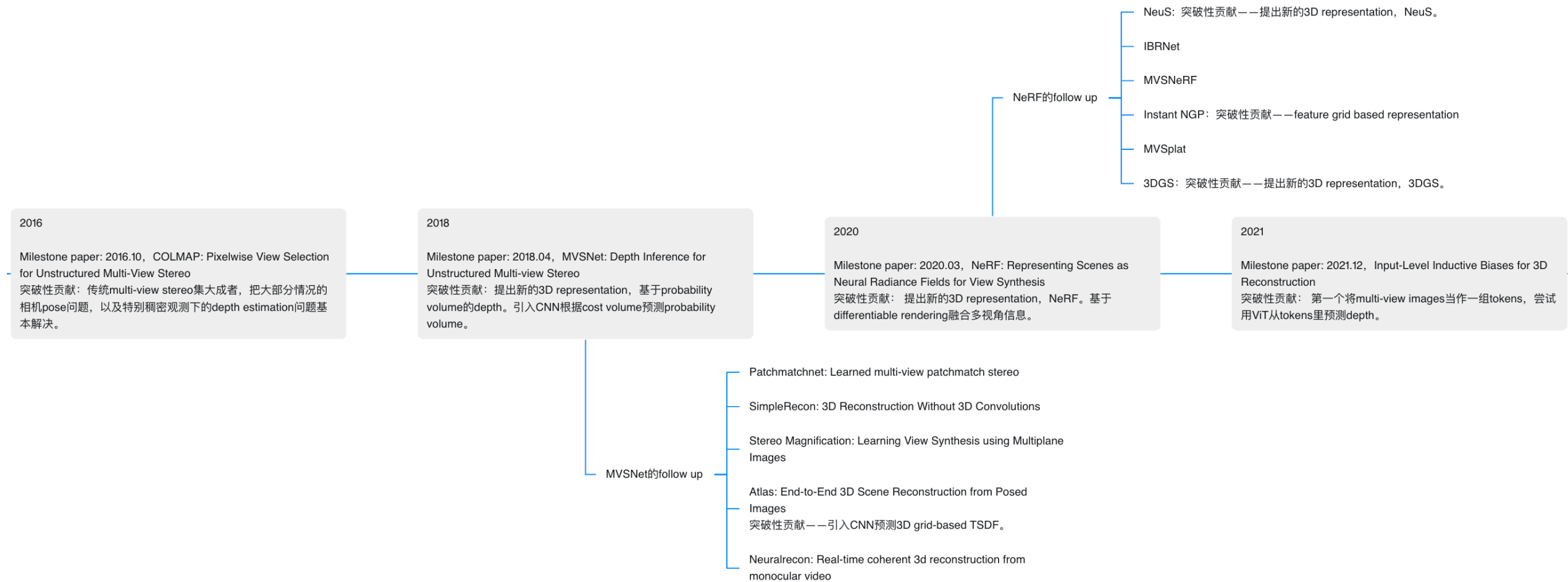
## 向后追踪

- 引用里程碑论文的论文
- 衍生工作



一个好用的工具: [Connected Papers](https://connectedpapers.org/)

# 怎么梳理技术发展脉络：时间轴思维导图





# 怎么梳理技术发展脉络

首先，初始化一个时间轴，将论文列到时间轴、并阅读每篇论文

- 了解它解决的问题、pipeline和技术见解。

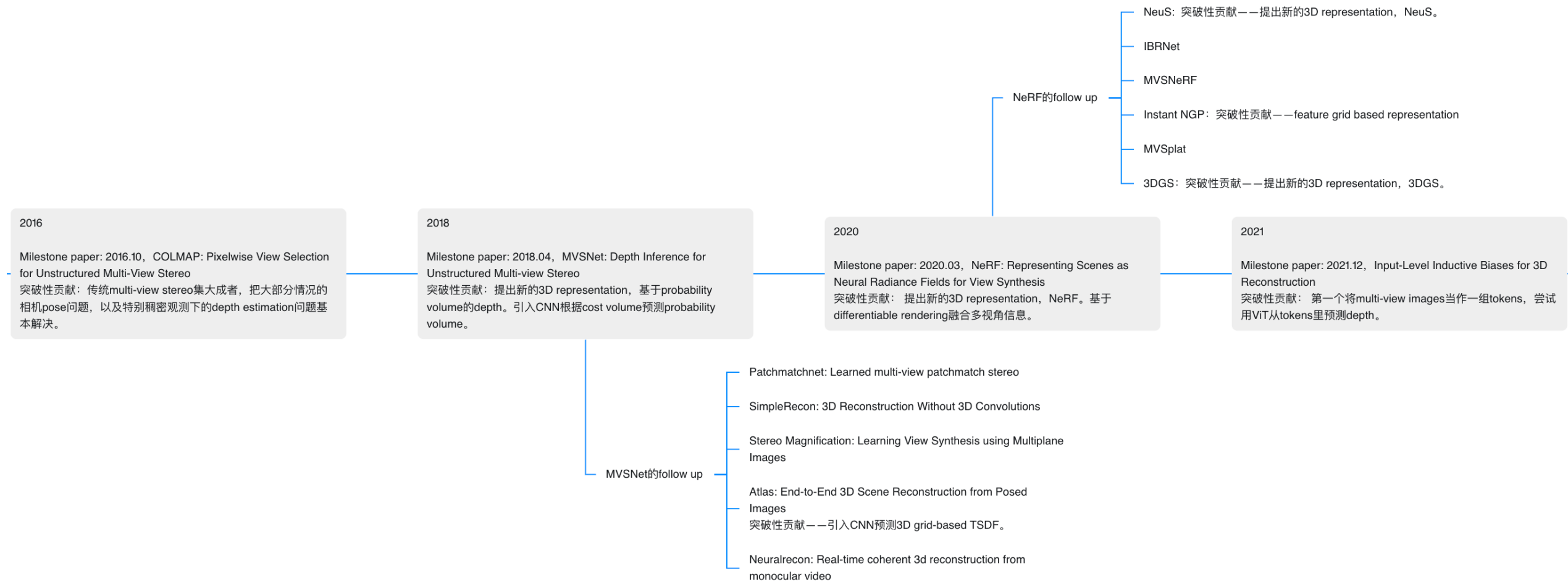
然后，确认哪些论文是milestone paper、哪些论文是follow-up

- 根据论文方法的创新性判断。

最后，总结这些论文

- Milestone paper的技术范式
- Follow-up papers作出的改进

# 怎么梳理技术发展脉络：时间轴思维导图



# 怎么梳理大家所关心问题的发展脉络

## 记录领域每年在解决的问题

- 根据构建的思维导图，看看每年的热点问题是什么。

## 总结该领域的发展

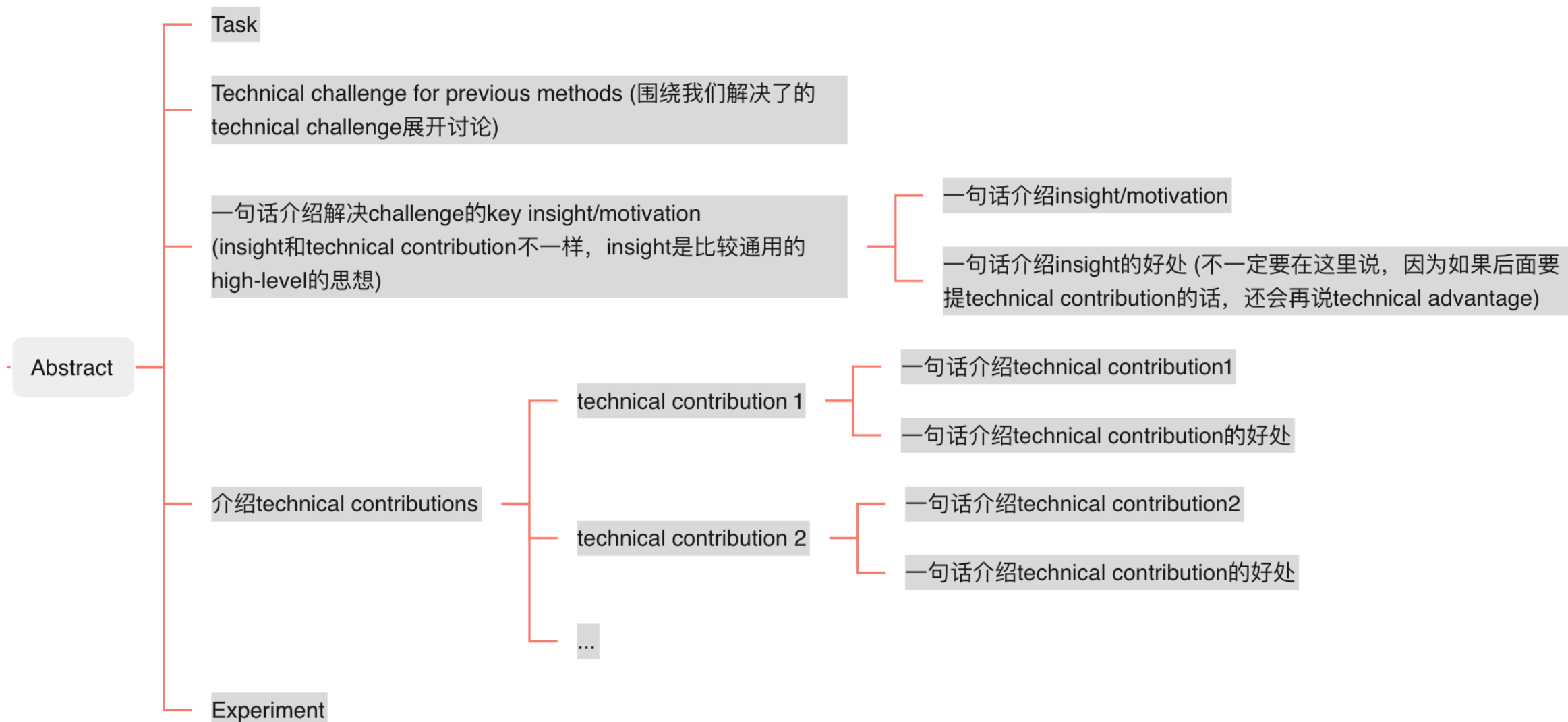
- 这个领域的终极目标是什么？
- 该领域已经达到了什么水平？
- 还有哪些重要的问题仍未被解决？
- 现阶段的热点话题是什么？

# 如何有效阅读论文

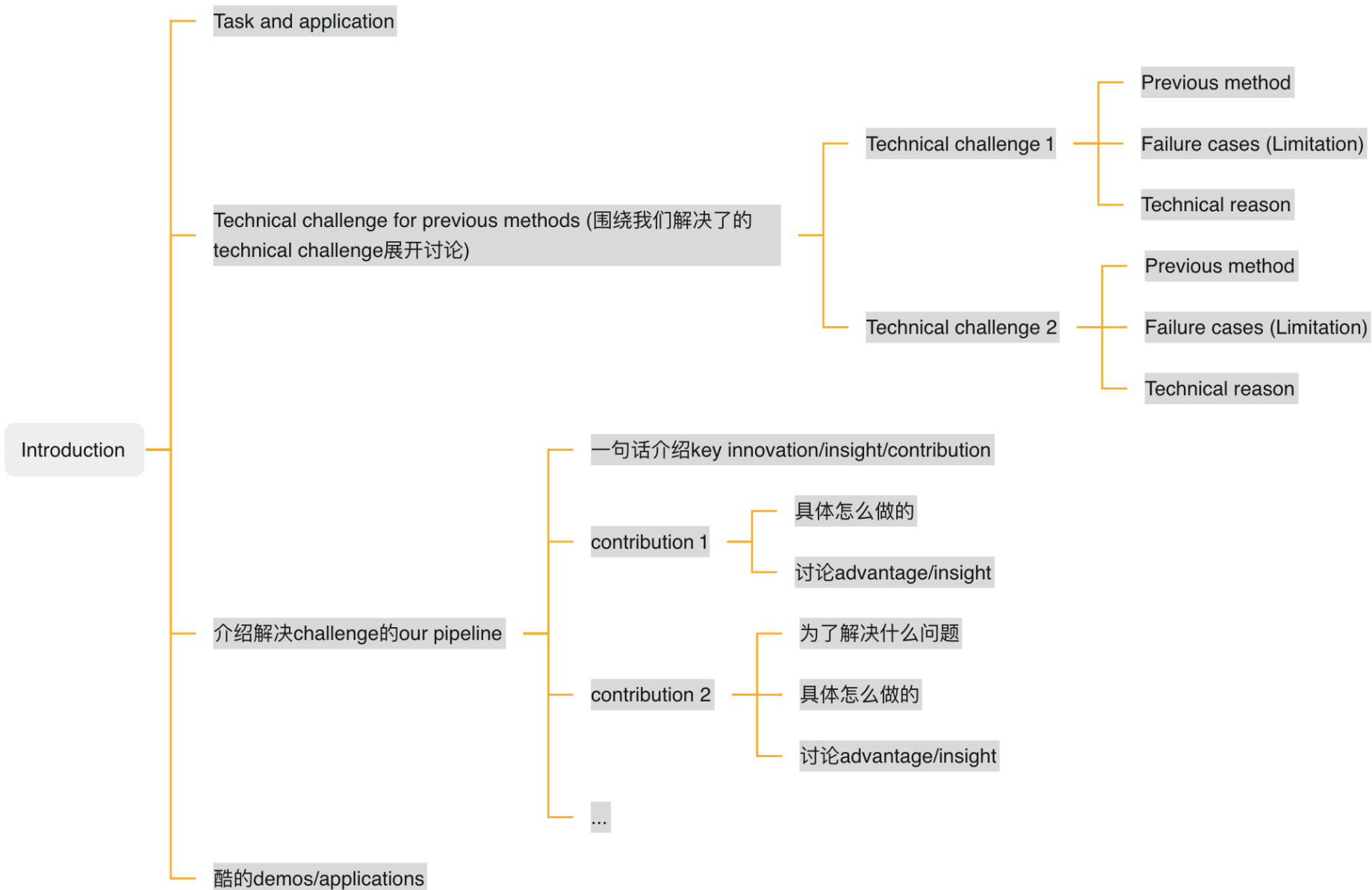
可以借助论文解析树来解决这个问题

- 将阅读论文的过程转变为回答问题的过程

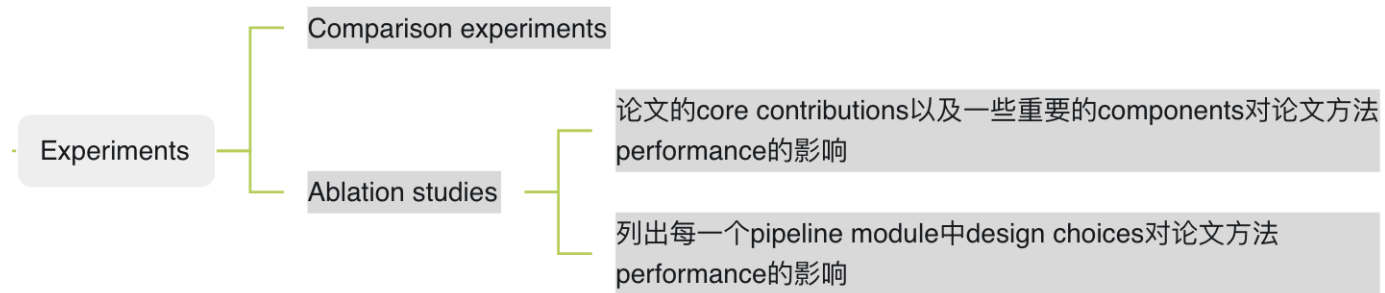
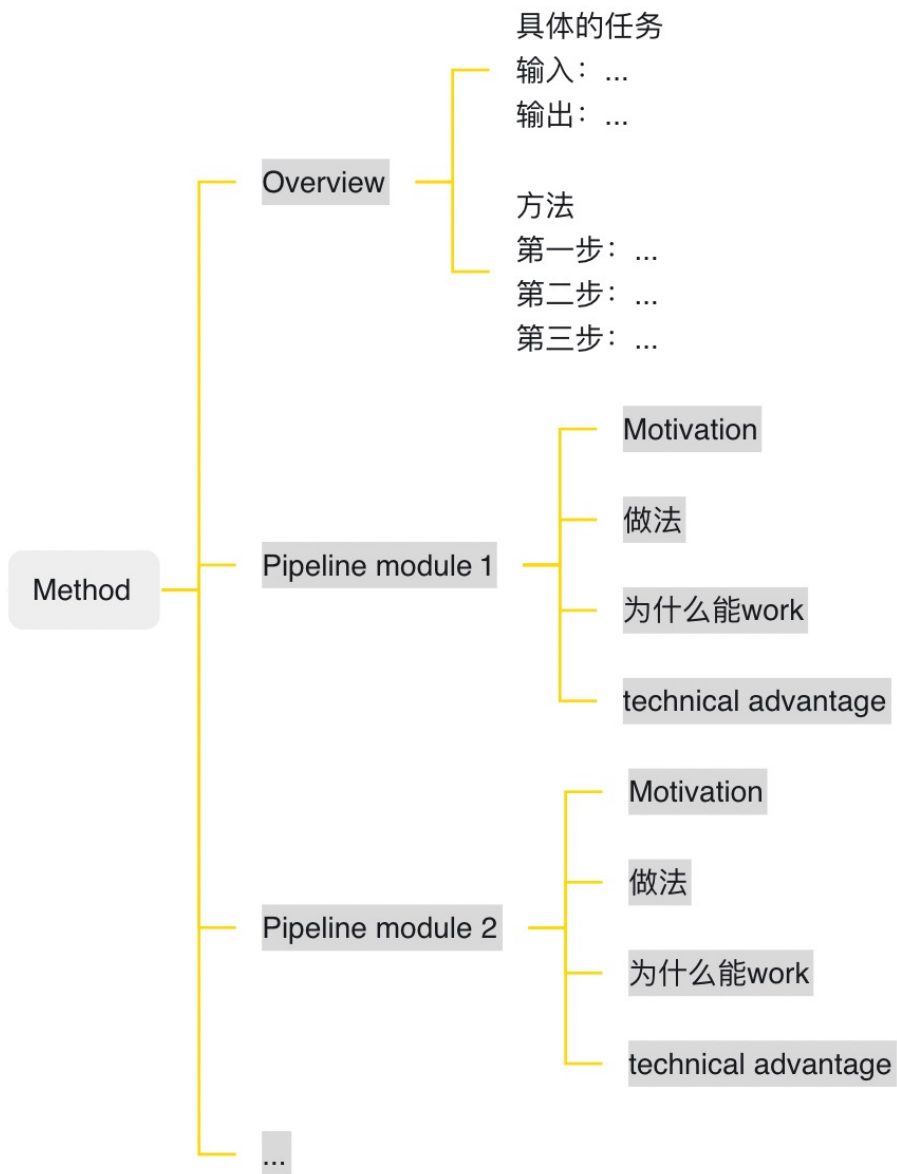
# 如何有效阅读论文



# 如何有效阅读论文



# 如何有效阅读论文



# 要建立多大的视野?



匹配个人能力



初学者往往只需建立较为视野narrow的视野



当一个人越来越强，看的论文越来越多，视野也越来越大



# 建立视野的额外方式

## 观看知名研究人员的演讲

- Noah Snavely (Planet-scale visual understanding <https://www.youtube.com/watch?v=UHkCa9-Z1Ps>)
- 3DGV <https://3dgv.github.io/>
- UW CS Colloquia <https://www.youtube.com/@uwcse/playlists>

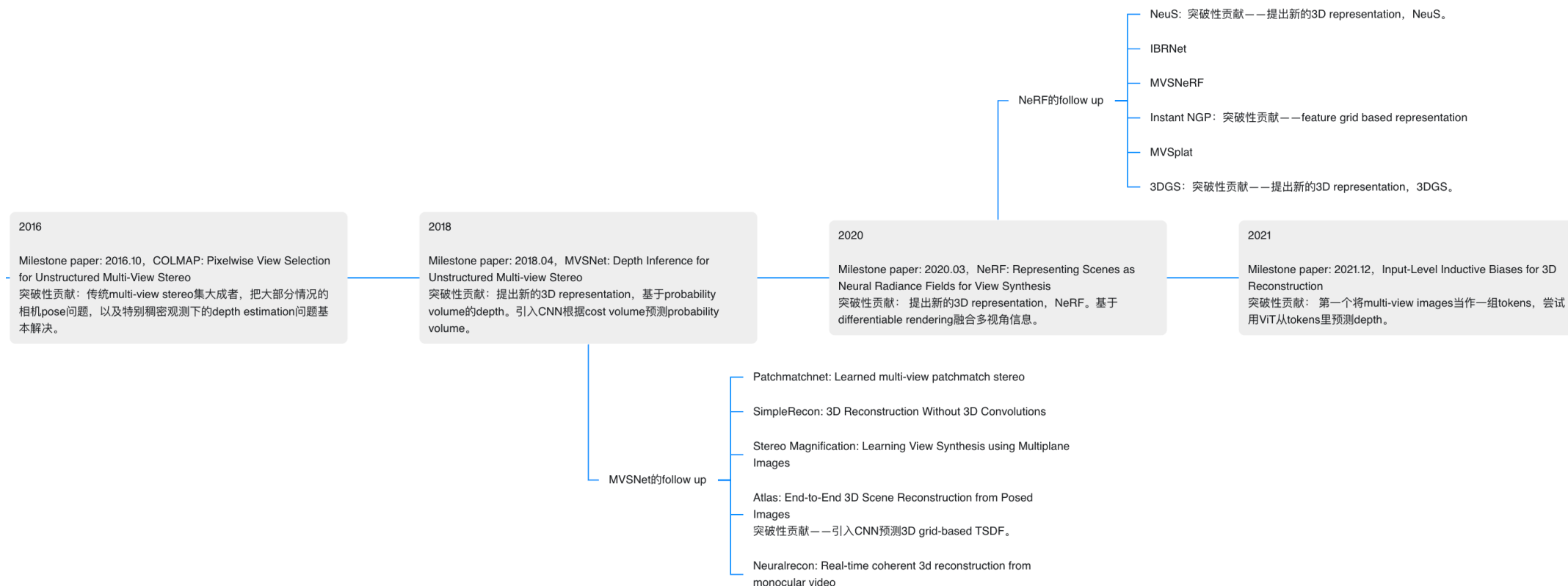
## 和他人交流

- 实验室伙伴、Senior advisors和知名研究人员

# 作业

- 整理一个在自己研究方向上的时间轴思维导图，并总结该方向当前时间点仍未解决的重要问题。
- 提交格式：PDF文件。

- 例子：





# QA：回答同学们的一些问题

- 入门时遇到的问题
- 作为一作做Project时遇到的问题

# QA：入门时遇到的问题

问题：最近刚开始读论文，感觉深入理解论文和代码有难度，有什么建议吗，有哪些学习技巧。

回答：

⚠️ 什么情况下，看代码的效率会很低：没有目的性地看代码。  
这种情况下，即使看完代码，清楚了逻辑，也不知道满足了自己的什么需求，导致没有收获。

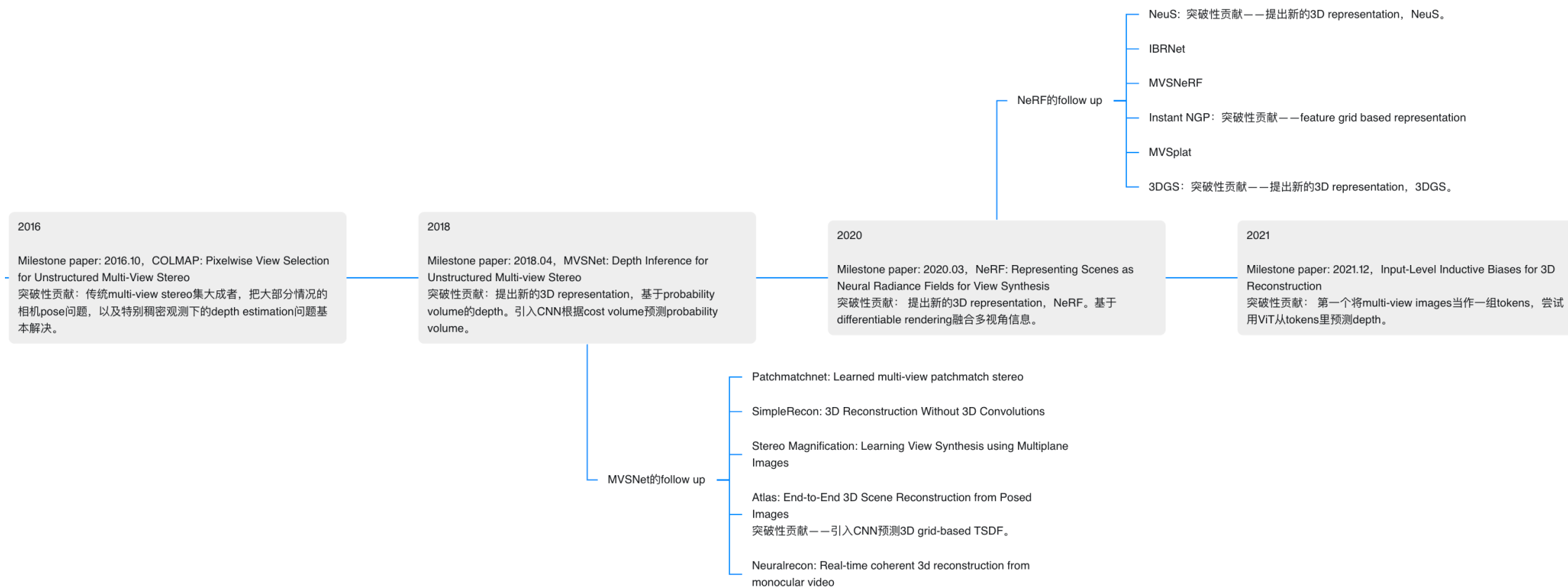
一种提升学习代码效率的技巧（**对入门者很好用**）：

1. 设置一个对论文算法的简单改进任务，并进行实现。
2. 这个任务会驱动我们去读懂相关的代码。因为只有读懂代码以后，才知道怎么改进。
3. 在实现算法改进的过程中，自己的算法编程能力也得到了提升。

# QA: 入门时遇到的问题

问题：如何能够快速有效地从零开始建立起对于一个领域工作的认知；针对于三维重建领域如何进行系统的学习。

回答：





## QA：入门时遇到的问题

问题：如果刚进入一个新的领域，没有太多相关基础，同时老师给的课题又有一定超前性（缺乏相关研究基础），这样的话在接下来很长的一段时间中最重要的一些事情或想法是什么？

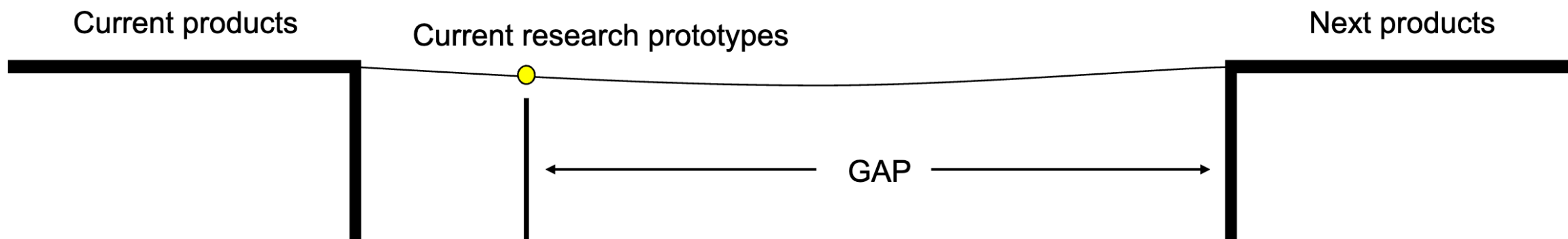
回答：上手一篇开源的baseline paper，设立小的改进任务以学习其算法与代码。

# QA: 作为一作做Project时遇到的问题 -- 选择研究方向

问题： 如何选择一个具有较长发展/探索前景的研究方向？

具有强烈个人色彩的回答，请酌情参考：

明确自己想实现的product，看看research与product存在多大的gap。



From Andrew Davison

# QA: 作为一作做Project时遇到的问题 -- 选择科研课题

问题： 如何选择一个新的课题？

回答：

## 寻找好课题的四个步骤

建立领域的视野

- 整理milestone papers/技术范式。
- 了解它们是如何随着时间演变的。

列出领域内的课题

- 列出该领域的重要问题。
- 判断哪个问题是当前时间点应该去解决的。

判断哪些课题是好课题

- 评估课题的竞争程度和发展空间。
- 考虑任务的难度与个人能力相匹配。

在选题上Aim High

- 考虑任务的影响力。
- 技术风险性大的课题往往有更高的收益。

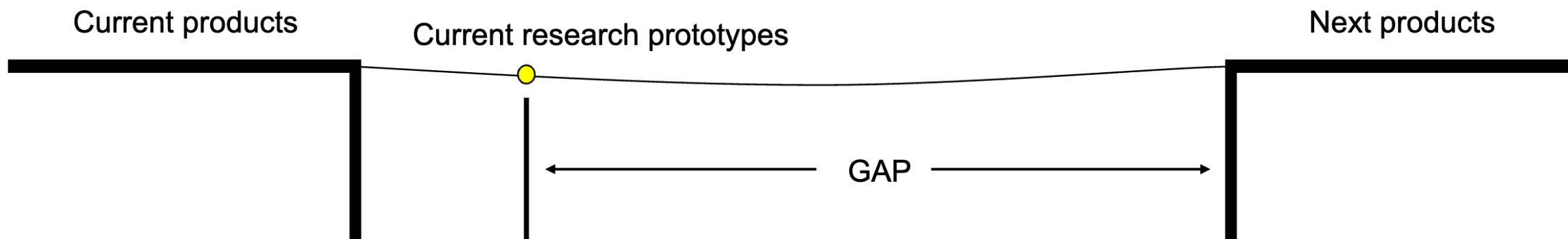


# QA: 作为一作做Project时遇到的问题 -- 选择科研课题

问题：请问怎样在实际工程中提炼科学问题？

回答：

1. 首先判断实际工程是否有较大的提升空间。



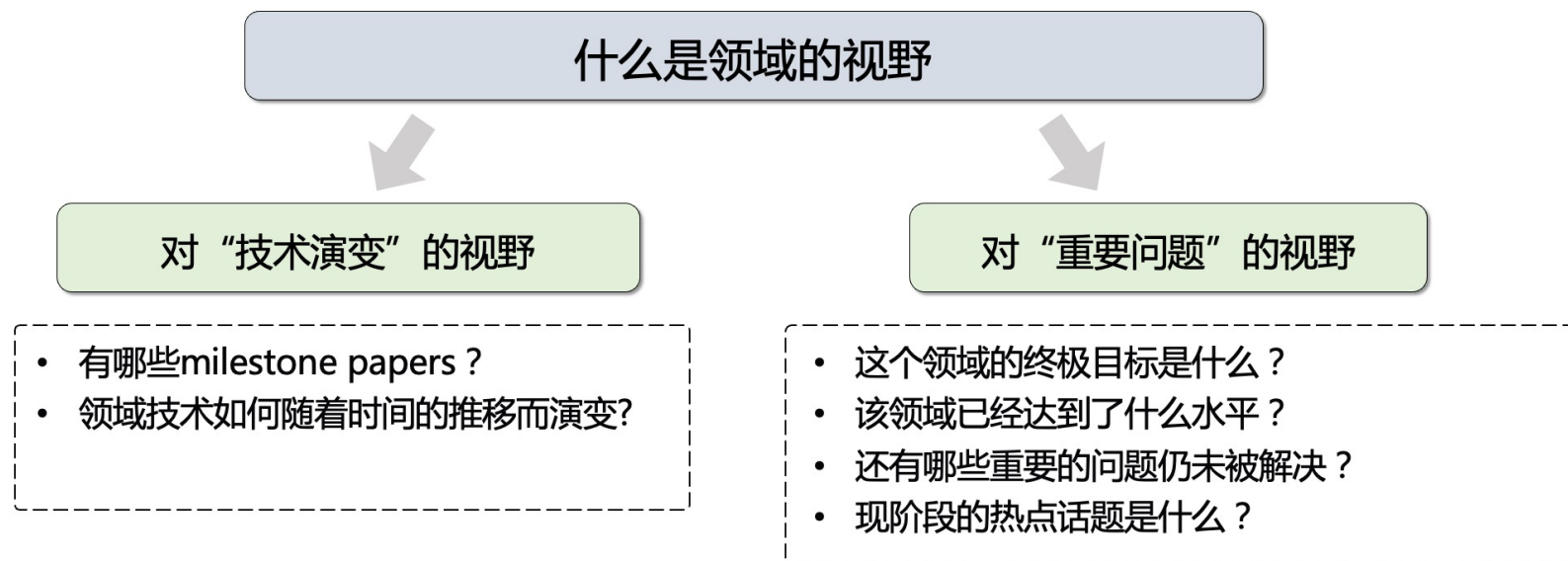
From Andrew Davison

# QA: 作为一作做Project时遇到的问题 -- 选择科研课题

问题：请问怎样在实际工程中提炼科学问题？

回答：

1. 首先判断实际工程是否有较大的提升空间。
2. 发现仍存在的重要问题。



## QA：作为一作做Project时遇到的问题 -- 选择科研课题

问题：请问怎样在实际工程中提炼科学问题？

回答：

1. 首先判断实际工程是否有较大的提升空间。
2. 发现仍存在的重要问题。
3. 以第一性原理分析该问题的本质原因。

## QA：作为一作做Project时遇到的问题 -- 选择科研课题

问题：如何快速判断一个课题是否值得深入研究？

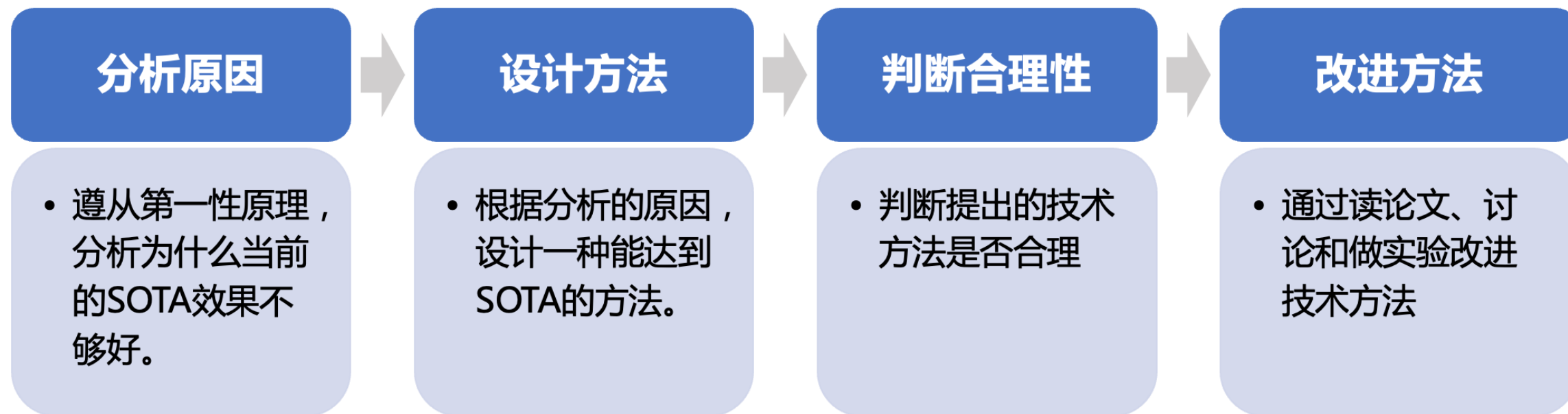
回答：

1. 依赖个人的“领域视野”与“实验经验”。
2. 可以多和人讨论。

# QA: 作为一作做Project时遇到的问题 -- 提出解决方案

问题：在解决问题的过程中，怎样确定具体解决方法？

回答：



## QA：作为一作做Project时遇到的问题 -- 提出解决方案

问题：如何从海量的技术路线中选择最为合适的，因为如果路线都探索一遍，比较费时间，如何能高效地做出选择？

回答：

1. 依赖个人的“领域视野”与“实验经验”。
2. 可以多和人讨论。

# Project时间规划

问题：

- 一篇工作的花费时间大概是怎样呢？
- 每部分投入的时间？
- 如何控制每部分的时间，什么时候能够结束这部分进入下一部分呢？

	具体应该完成的事情	预期完成时间
目标会议		CVPR (2024年11月15日)
节点6 (做论文, 完成投稿)	<ol style="list-style-type: none"><li>1. 梳理论文的story (倒推, 然后正推。详细做法见Introduction部分)。整理成story以后, 写一个introduction的思路。同时列出要做的comparison experiments和ablation studies, 并开始做起来。先做ablation studies, 确定最优的方法版本, 再做对比实验。Baseline methods的实验可以同步跑起来。</li><li>2. 画一个清楚的pipeline figure的草图 (流程步骤定清楚)。</li><li>3. 写method, 同时做实验。</li><li>4. 改introduction和method, 同时做实验。</li><li>5. 实验做差不多以后, 写experiment。</li><li>6. 写related work。</li><li>7. Review论文。改论文的introduction、method和experiment。</li><li>8. 写abstract, 取论文名字。</li><li>9. 反复review论文, 改论文。</li></ol>	2024年11月15日
节点5 (很好地展示论文吸引人的点, 尽可能地扩大论文的影响力)	<ol style="list-style-type: none"><li>1. 规划清楚论文要做的demo、application</li><li>2. 用SGD process提升idea, 把demo、application的效果做好。</li></ol>	2024年9月30日
节点4 (设计验证性实验, 初步验证 technical idea, 改进idea)	SGD process: <ol style="list-style-type: none"><li>1. 想出一个Idea。</li><li>2. 通过文献阅读、与人讨论的方式提升Idea。</li><li>3. 实现Idea。</li><li>4. 设计探索性实验, 逐步验证idea。</li><li>5. 分析实验结果, 获得gradient来提升idea。</li></ol>	2024年8月31日
节点3 (基于第一性原理设计解决failure case的technical idea)	<ol style="list-style-type: none"><li>1. 分析sota不work的核心的技术原因、技术难点。</li><li>2. 列出所有的、有这个技术难点相关的, 可能可以解决该难点的techniques (脑图)。</li><li>3. 每天冥思苦想, 排列组合出大量的pipelines。</li><li>4. 和身边的大量人讨论。</li><li>5. 等待一个灵感迸发的契机, 从大脑子涌现出最work、最有新意的pipeline。</li></ol>	2024年6月31日
节点2 (明确想解决的failure case和 technical challenge。寻找能启发技术贡献的failure case)	在新的数据、新的setting下跑sota代码, 发现failure case。 <b>寻找能启发技术贡献的failure case。</b> 发现一个任务的technical challenge。回答下面的问题: <ol style="list-style-type: none"><li>1. 这个任务的setting有什么独特之处?</li><li>2. 这个任务的输入数据有什么独特之处?</li><li>3. 这个任务的输出有什么独特之处?</li></ol>	2024年6月15日
节点1 (规划project task)	<ol style="list-style-type: none"><li>1. 建立对该研究方向的视野。</li><li>2. 思考该研究方向目前sota还没完成哪些指标。</li><li>3. 列出可做的project tasks, 思考各个project task的竞争程度和发展程度, 选择符合个人能力的project task。</li></ol>	2024年5月31日
开始project的时间		2024年5月25日



# Panel Discussion

1. 这门课程希望通过什么方式帮助想做科研的同学入门？
2. 入门时大家遇到了什么困难，怎么解决的？
3. 做一个research project时陷入困难、停滞不前时，怎么解决的？
4. 大家什么时候开始感觉自己科研能力迅速提升的？
5. 为什么选择现在这个研究方向？
6. 是否应该为了好找工作，选择工业界目前火热的方向？
7. 做科研的动力是什么？如何找到做科研的动力/热情？





# Thank you!

---



彭思达



高俊



彭崧猷



王倩倩