



GAMES 003 科研素养课

第四周：面向技术方案的实验设计



Sida Peng



Jun Gao



Songyou Peng



Qianqian Wang

从伽利略落体实验谈起

第五节 自由落体运动



1590年，伽利略在比萨斜塔上做了“两个铁球同时落地”的著名实验，从此推翻了亚里斯多德“物体下落速度和重量成比例”的学说，纠正了这个持续了1900年之久的错误结论。

亚里士多德观点：

- 物体下落速度与物体重量成正比

伽利略观点：

- （不受空气阻力情况下）物体下落速度与物体重量无关

从伽利略落体实验谈起

- 需要验证的假说：
 - 物体下落速度与物体重量无关
- 实验设定：
 - 两个小球从高处下落
 - 而不是：两辆车从1米高下落
- 控制变量：
 - 两个质量不同的小球在相同位置相同时间下落
 - 而不是：
 - 两个质量不同的小球，一个在比萨，一个在北京下落
 - 两个质量不同的小球一个在白天，一个在晚上下落

第五节 自由落体运动



1590年，伽利略在比萨斜塔上做了“两个铁球同时落地”的著名实验，从此推翻了亚里斯多德“物体下落速度和重量成比例”的学说，纠正了这个持续了1900年之久的错误结论。

什么是实验设计?

实验 [编辑]

条目 讨论 大陆简体 ▼

此条目可参照[英语维基百科相应条目](#)来扩充。

A → **文** 若您熟悉来源语言和主题，请协助参考外语维基百科扩充条目。请勿直接提交机械翻译，内容。依版权协议，译文需在编辑摘要注明来源，或于讨论页顶部标记 `{{Translated`

实验（英语：experiment）是在**设定的条件**下，用来检验某种假设，或者验证或质疑某种已经存在的理论而进行的操作。通过**控制变量**产生不同的结果，实验有助于人们理解现象之间的因果关系。实验的目标各有不同，尺度也有大有小，但实验总是依赖**可重复**的操作，以及**对结果的逻辑分析**。



课程内容

- 实验设计
 - 分解问题到最简化的实验来探索/验证技术方案
- 控制变量
 - 分析实验结果，以改进技术方案
- 可重复实验
- 实验执行
- 如何面对实验失败

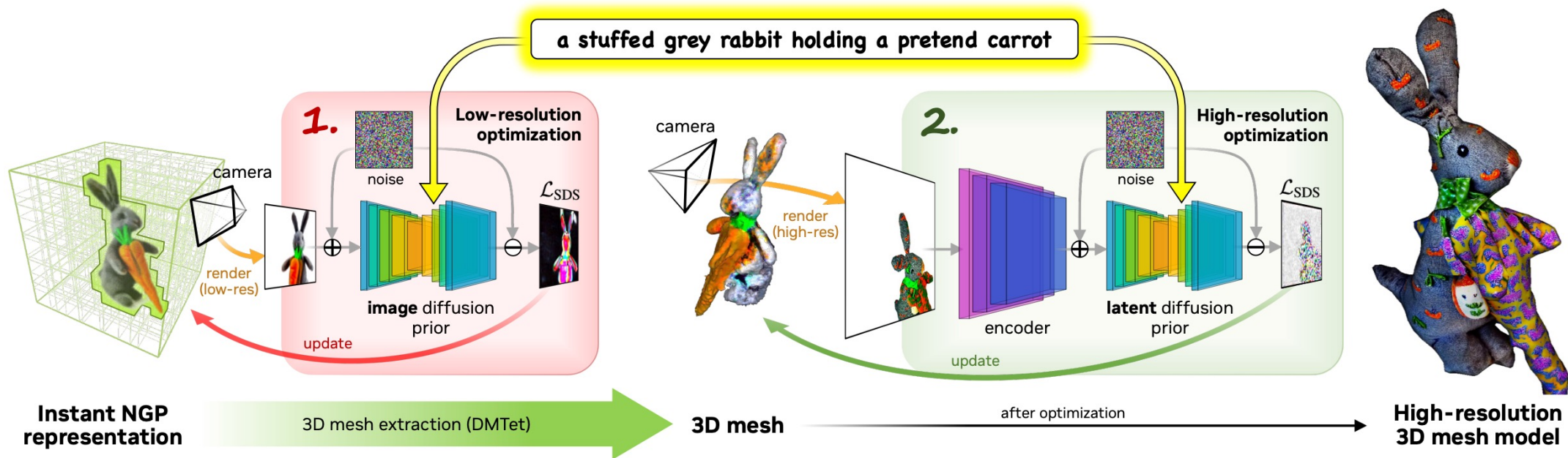


实验设计的出发点

- 假设检验
 - 验证我们的想法 (idea) 是否有用?
- 怎么样去验证 “我们的想法是否有用” ?
 - 我们具体有哪些假设 (想法) ?
 - 控制变量来帮助我们做逻辑分析。

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分



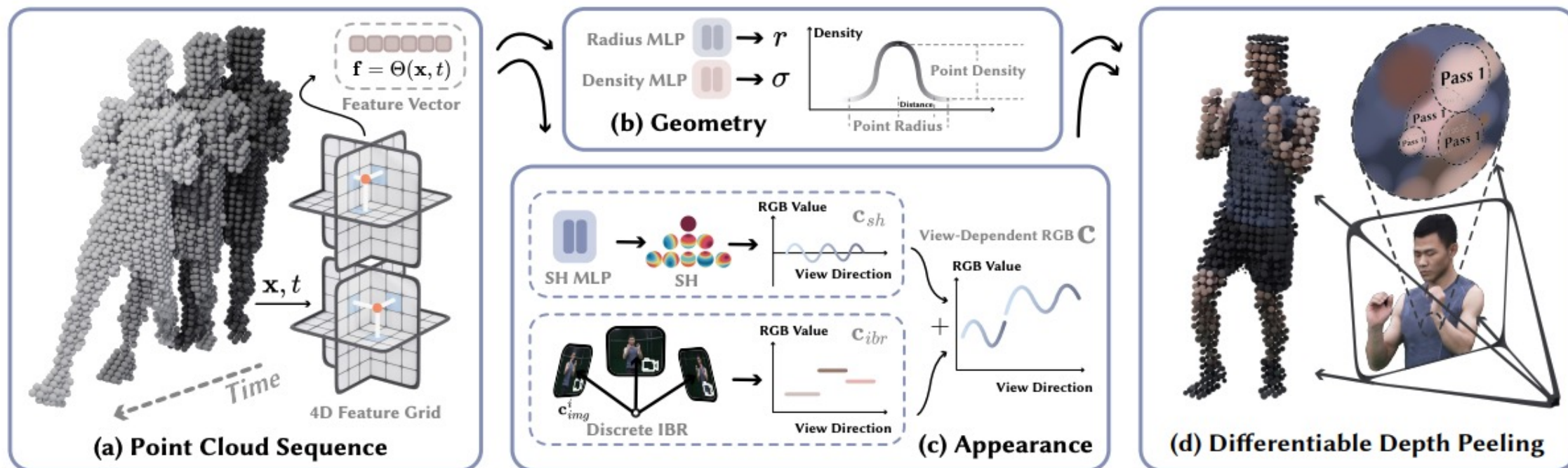
问题拆解:

1. Implicit representation (assume low-res diffusion prior)
2. Explicit representation (assume a good initialization)
3. Diffusion prior (low-res v.s. high-res.)

Magic3D

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分



问题拆解:

1. 4D feature grid
2. Geometry modeling
3. Appearance modeling
4. Depth peeling

4K4D
Credit: Sida Peng

<https://zju3dv.github.io/4k4d/>

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
 - 怎么拆解大的方法?

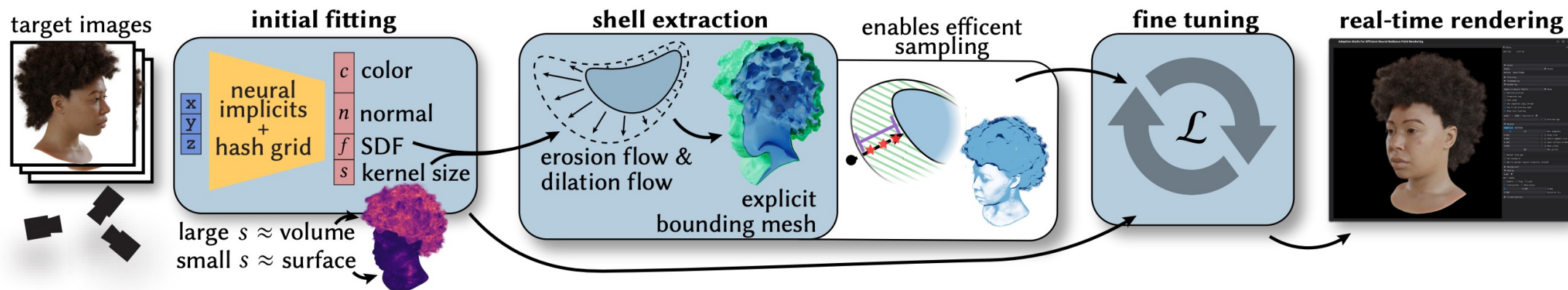


Fig. 3. Overview of the proposed approach. We demonstrate high-fidelity, efficient neural implicit scene reconstruction by efficiently-sampling volumetric rendering inside of an explicit thin shell, which is automatically fit from visual objectives.

把方法的流程图画出来 -> 画图的过程就是在帮我们看清楚整体的框架

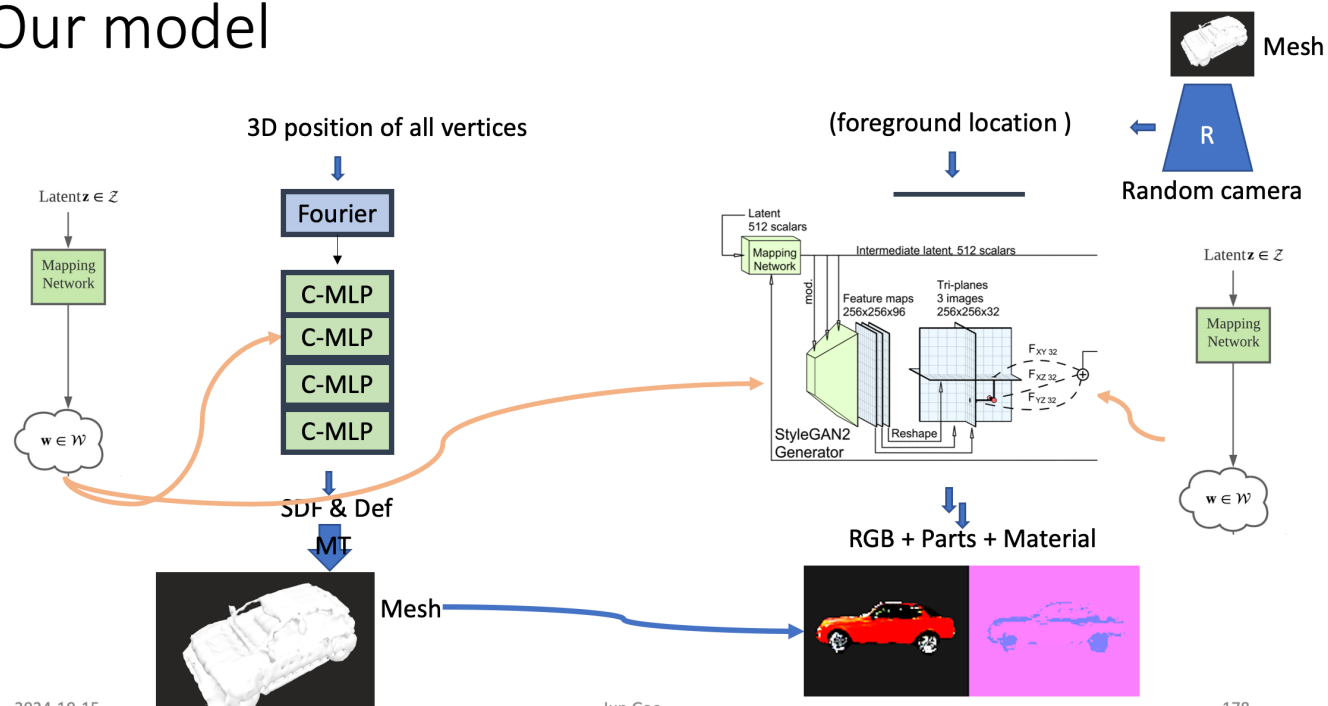
AdaptiveShell

<https://research.nvidia.com/labs/toronto-ai/adaptive-shells/>

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
 - 怎么拆解大的方法?

Our model



GET3D草图

流程图并不一定需要像paper一样完美, 目的是为了整理自己的思路

我们到底有哪些想法？

- 把一个大的方法分解成许多个小的部分
 - 怎么拆解大的方法？
 - 能提前预知好完美的拆解办法吗？
 - 比如把大象放进冰箱，一般大家会分解成“打开冰箱，放进大象，关上冰箱”，但也有可能有“找一个大象，找一个冰箱，打开冰箱，尝试放入大象，压缩大象，循环到放入大象成功，关上冰箱”。
 - **边做实验边优化自己的分解方案。**先得讨论出一个大致正确的分解框架，然后开始实验。
 - 比如：当打开冰箱的时候发现没有冰箱，那就需要先找到冰箱，但万一已经有了冰箱呢？
 - 比如：当发现直接放大象放不进去的时候，那就需要压缩大象，但万一直接放进去了呢？

Q&A

- 做实验时是先做整个pipeline的实验再做每个部分的消融还是先验证每个部分的有效性再做整个pipeline的实验?
 - 建议：可以先搭好一个最基本的pipeline，然后再去验证每一个部分
 - 最基本的pipeline意味着每一部分可以很简单
 - 搭好pipeline让我们对宏观上的实验有更好的把握（避免拆分得不完美）
 - 也可以验证每一部分放到pipeline里面对最终结果的影响（避免没法组合到一块）

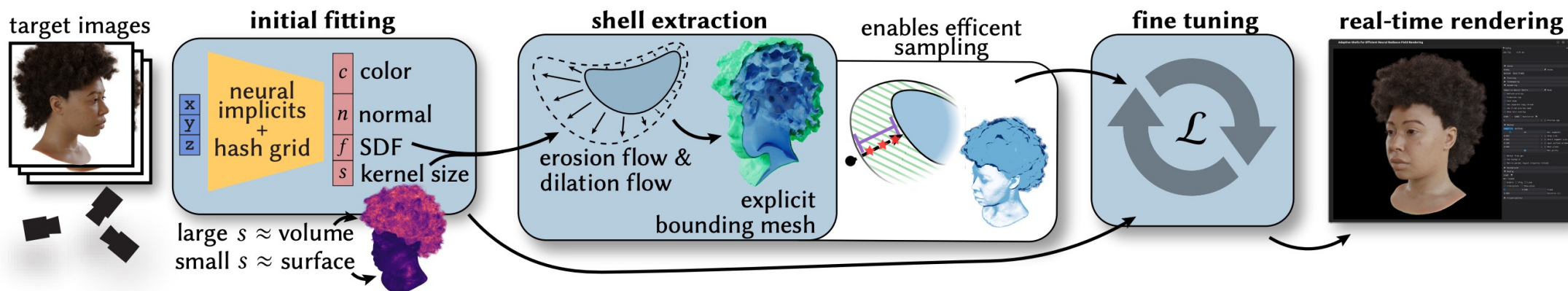


Fig. 3. Overview of the proposed approach. We demonstrate high-fidelity, efficient neural implicit scene reconstruction by efficiently-sampling volumetric rendering inside of an explicit thin shell, which is automatically fit from visual objectives.

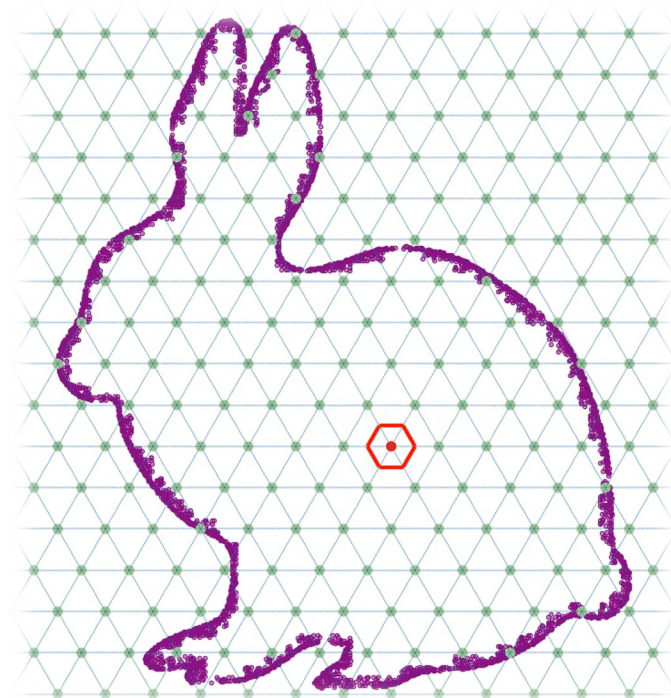
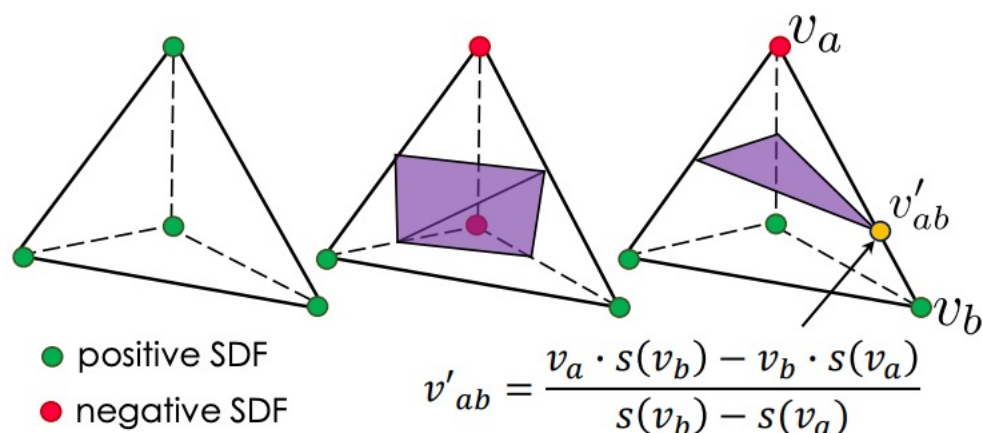


Q&A

- 把大问题拆解成小问题 V.S. end-to-end
 - End-to-end 本身就是一个大问题
 - E.g. 该选用什么的data, 什么样的loss function, 怎么end-to-end training
 - 澄清:
 - 并不是说解决科研问题我们一定需要把大问题转化成小问题, 很多科研问题本身需要end-to-end
 - 而是对于我们的技术方案进行拆分, 把一个大的技术方案拆分成每一个可以独立验证的部分去分别研究。

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题, 从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting

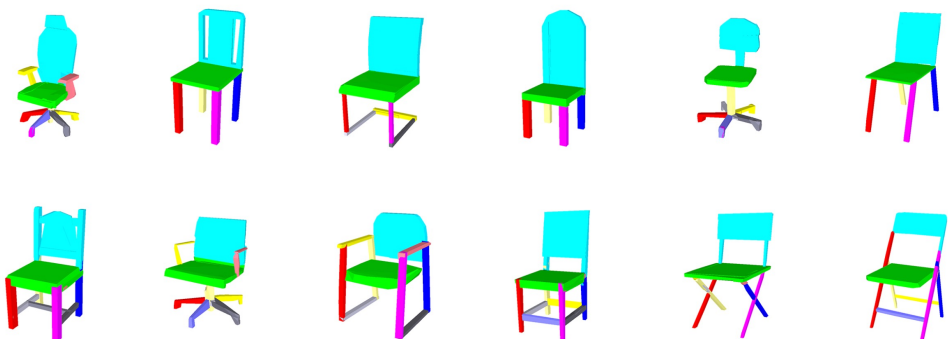


验证Marching Tetrahedra操作是否可导

举例: 研究3D的问题, 可不可以先从2D开始?

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题, 从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting



验证是否可以用BSP- tree来表示三维物体

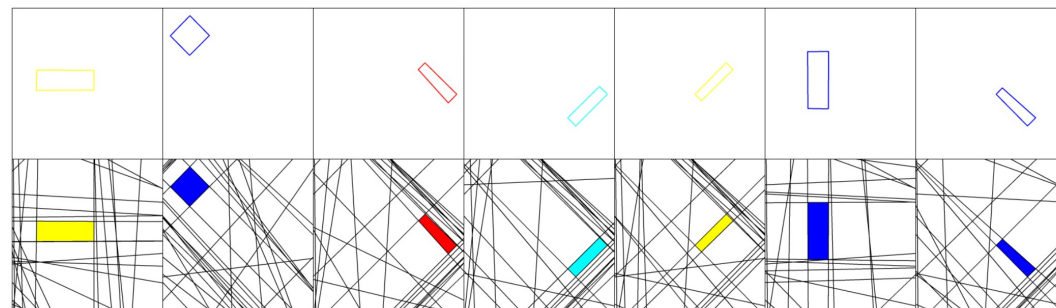
In our toy 2D experiment,

the task of the network is to reconstruct each image as a combination of convex parts.

Since all shapes share the same set of convexes and tree connections,

we find shape segmentations and correspondences in the convex level.

Below, we show a few convexes from the first shape, and the planes to construct them.



举例: 研究3D的问题, 可不可以先从2D开始?

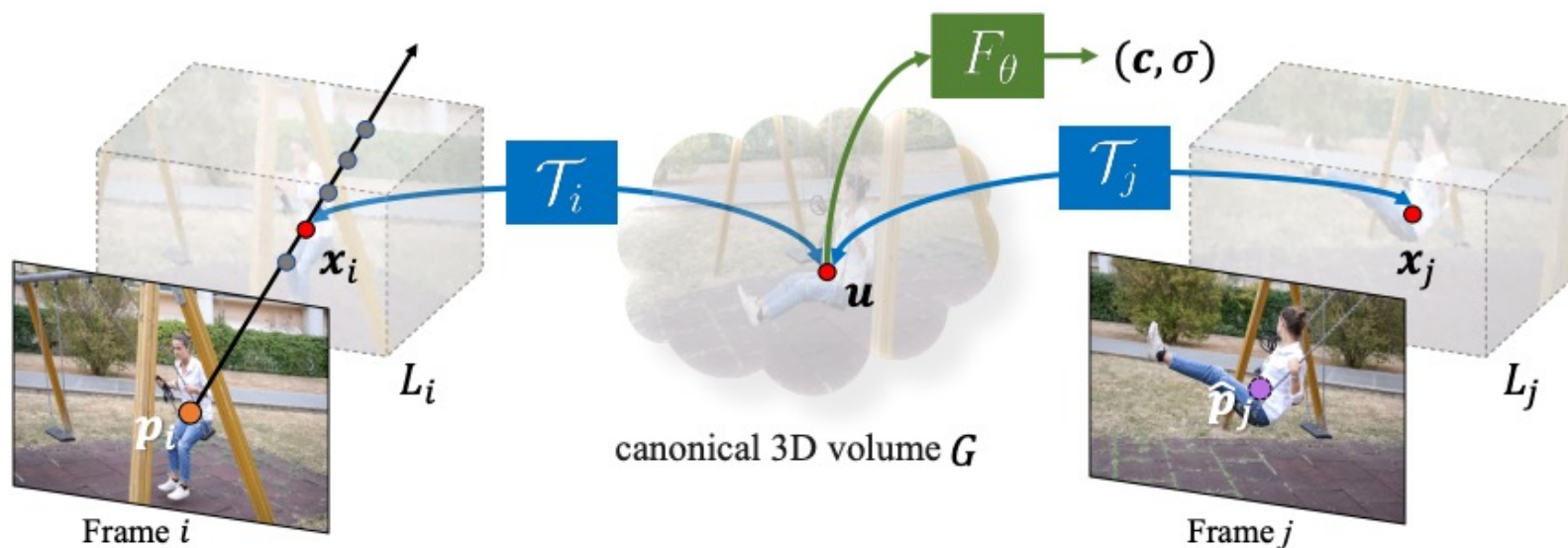
BSPNet

Credit: Zhiqin Chen

<https://bsp-net.github.io/>

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题，从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting
 - 简化实验的数据 (比如: 数据有噪声, 相机参数不准确)

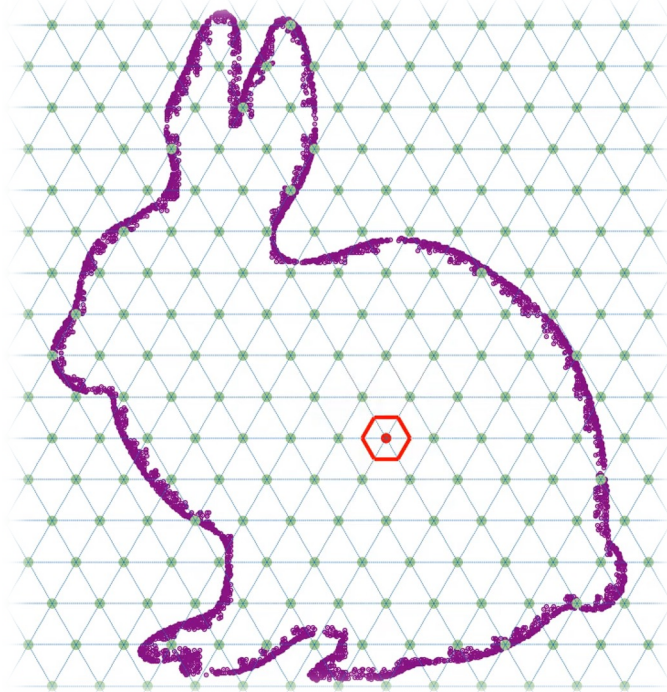


(a) OmniMotion representation

验证Invertiable Network是否可以表达长距离的correspondence

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题, 从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting
 - 简化实验的数据 (比如: 数据有噪声, 相机参数不准确)



验证Marching Tetrahedra操作是否可导

举例: 假设我们有perfect ground truth

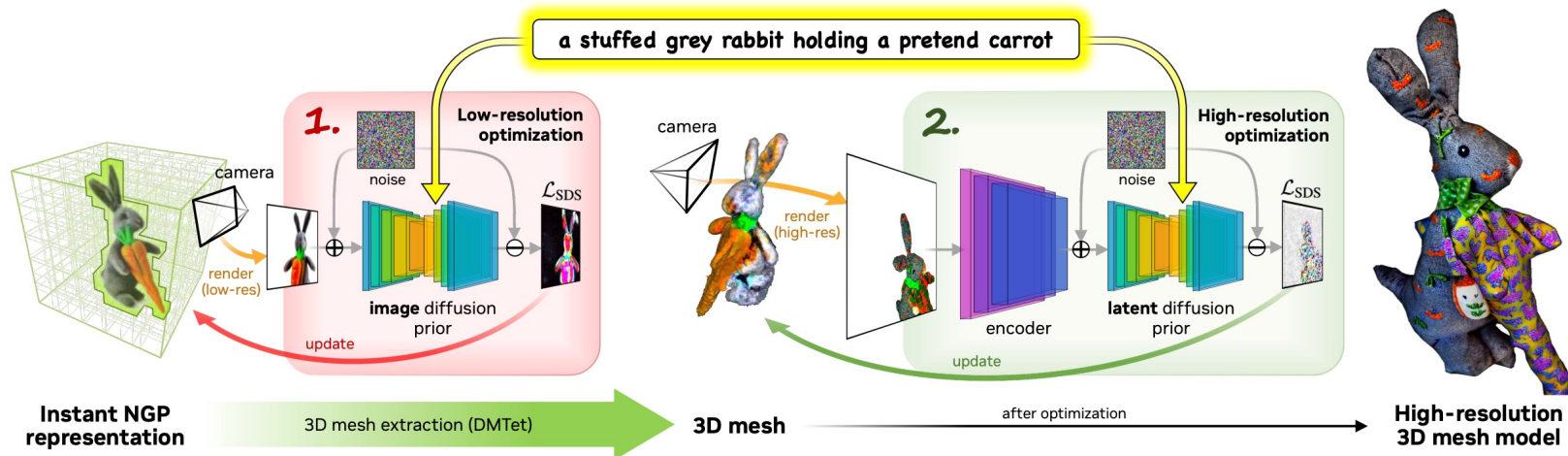
我们到底有哪些想法？

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题，从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting
 - 简化实验的数据 (比如：数据有噪声，相机参数不准确)

举例：在训练神经网络之前先优化一个例子 (确定这个神经网络能overfit) 。

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题，从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting
 - 简化实验的数据
 - 排除其他因素的干扰，假设这个系统里面别的模块都完美



问题拆解:

1. Implicit representation (assume low-res diffusion prior)
2. Explicit representation (assume a good initialization)
3. Diffusion prior (low-res v.s. high-res.)

我们到底有哪些想法?

- 把一个大的方法分解成许多个小的部分
- 简化每一个小问题，从最简单的且能验证技术方案的实验开始 (Toy example)
 - 简化实验的setting
 - 简化实验的数据
 - 排除其他因素的干扰，假设这个系统里面别的模块都完美
- 简化并不意味着忽略原本的问题
 - 简化是原本问题的提炼 (帮助我们理解原本的问题, debug code)
 - 从简化得到的方法需要重新被用到原本的问题 (原本的实验数据不能被简化)



课程内容

- 实验设计
 - 分解问题到最简化的实验来探索/验证技术方案
- 控制变量
 - 分析实验结果，以改进技术方案
- 可重复实验
- 实验执行
- 如何面对实验失败



控制变量

- 为什么我们要控制变量？
 - 帮助我们对实验结果进行逻辑分析，从而得到正确的结论
 - 如果同时有两个变量在影响结果，很难得到结论
 - 举例：
 - 两个质量不同的小球，一个比萨，一个在北京，下落，如果下落时间不同，我们无法判断是因为地理位置还是质量影响了实验结果。

控制变量

- 为什么我们要控制变量?
 - 帮助我们对实验结果进行逻辑分析, 从而得到正确的结论

有两个因素影响结果:

1. 更高resolution的diffusion model
2. 选择用DMTet来表示3D geometry

实验对比:

- A. Nerf + low-res diffusion model
- B. Nerf + high-res diffusion model
- C. DMTet + high-res diffusion model
- D. (DMTet + low-res diffusion model)



控制变量

- 为什么我们要控制变量？
 - 帮助我们对实验结果进行逻辑分析，从而得到正确的结论
- 做research的过程中很大程度上是在做Ablation study
 - 分析哪个地方做得不好，从而improve它

Table 4. Ablating our method on the *Shelly* data set. SV Kernel denotes the spatially varying kernel as introduced in Section 3.2. *Band, fixed* denotes the shell is not adaptive but extracted for a given SDF threshold.

Model	PSNR \uparrow	LPIPS \downarrow	SSIM \uparrow	Sample \downarrow
Ours (full ray, w/o SV kernel)	32.99	0.115	0.921	384
Ours (full ray)	34.26	0.104	0.932	384
Ours (band, fixed ± 0.05)	33.83	0.110	0.928	4.51
Ours (band, fixed ± 0.02)	31.14	0.136	0.913	2.29
Ours (keep regularization)	34.22	0.085	0.948	1.74
Ours	36.02	0.079	0.954	1.74

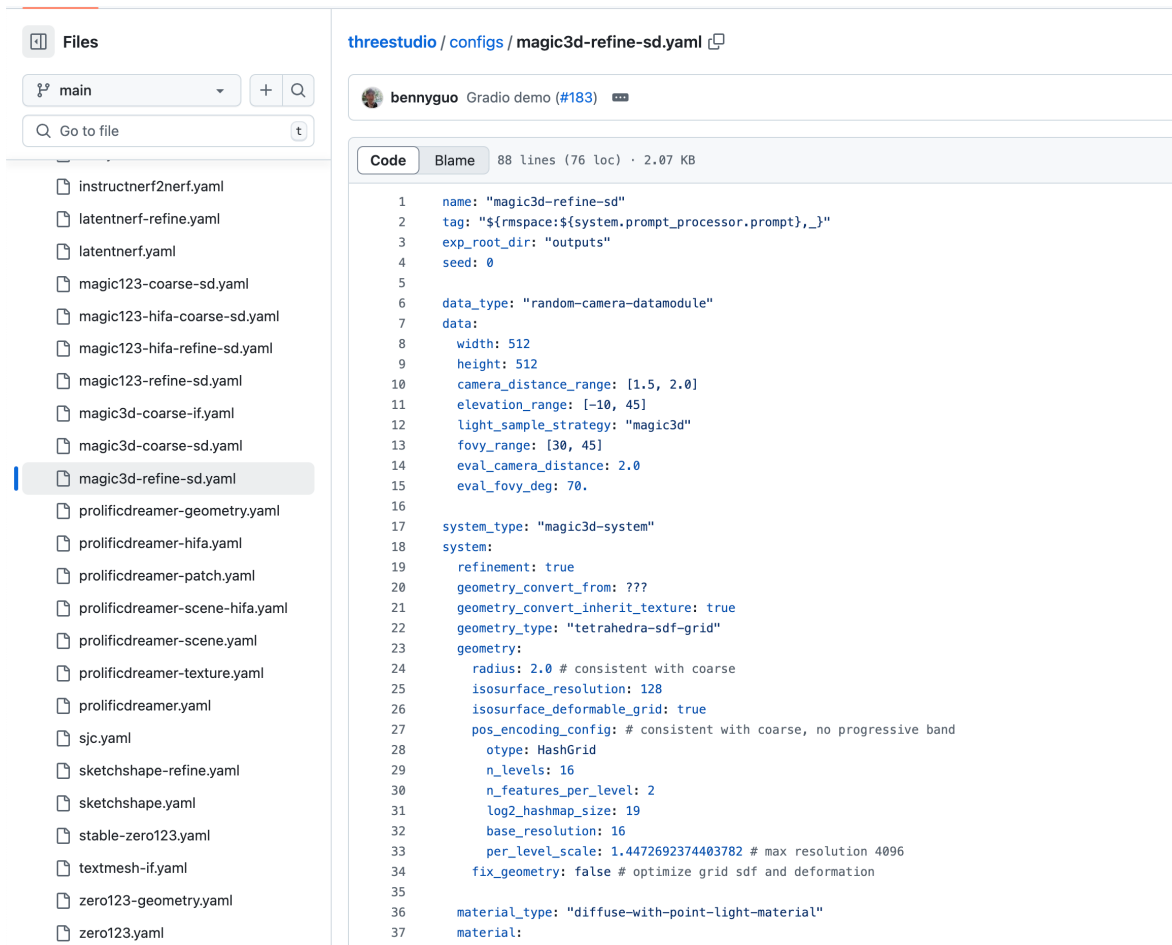
控制变量

- 为什么我们要控制变量？
 - 帮助我们对实验结果进行逻辑分析，从而得到正确的结论
- 做research的过程中很大程度上是在做Ablation study
 - 分析哪个地方做得不好，从而improve它
- 如何控制变量？
 - 保证实验过程中只有一个因素不一样
 - 对coding的需求（只改一个option就能实现不同的experiment settings）

```
173     ### Configs for 3D generator#####
174     @click.option('--iso_surface', help='Differentiable iso-surfacing method', type=click.Choice(['dm Tet', 'flexicubes']), default='dm Tet')
175     @click.option('--use_style_mixing', help='whether use style mixing for generation during inference', metavar='BOOL', type=bool, default=True, show_default=False)
176     @click.option('--one_3d_generator', help='whether we detach the gradient for empty object', metavar='BOOL', type=bool, default=True, show_default=True)
177     @click.option('--dm Tet_scale', help='Scale for the dimension of dm Tet', metavar='FLOAT', type=click.FloatRange(min=0, max=10.0), default=1.0, show_default=True)
178     @click.option('--n_implicit_layer', help='Number of Implicit FC layer for XYZPlaneTex model', metavar='INT', type=click.IntRange(min=1), default=1)
179     @click.option('--feat_channel', help='Feature channel for TORGB layer', metavar='INT', type=click.IntRange(min=0), default=16)
180     @click.option('--mlp_latent_channel', help='mlp_latent_channel for XYZPlaneTex network', metavar='INT', type=click.IntRange(min=8), default=32)
181     @click.option('--deformation_multiplier', help='Multiplier for the predicted deformation', metavar='FLOAT', type=click.FloatRange(min=1.0), default=1.0, required=False)
182     @click.option('--tri_plane_resolution', help='The resolution for tri plane', metavar='INT', type=click.IntRange(min=1), default=256)
183     @click.option('--n_views', help='number of views when training generator', metavar='INT', type=click.IntRange(min=1), default=1)
184     @click.option('--use_tri_plane', help='Whether use tri plane representation', metavar='BOOL', type=bool, default=True, show_default=True)
185     @click.option('--tet_res', help='Resolution for tetrahedron', metavar='INT', type=click.IntRange(min=1), default=90)
186     @click.option('--latent_dim', help='Dimension for latent code', metavar='INT', type=click.IntRange(min=1), default=512)
187     @click.option('--geometry_type', help='The type of geometry generator', type=str, default='conv3d', show_default=True)
188     @click.option('--render_type', help='Type of renderer we used', metavar='STR', type=click.Choice(['neural_render', 'spherical_gaussian']), default='neural_render', show_d
```

Q&A

能否推荐一些符合科研需求（比如简化参数配置便于迭代等等）的工具（第三方库）



```
161 # parse YAML config to OmegaConf
162 cfg: ExperimentConfig
163 cfg = load_config(args.config, cli_args=extras, n_gpus=n_gpus)
164
165 # set a different seed for each device
166 pl.seed_everything(cfg.seed + get_rank(), workers=True)
167
168 dm = threestudio.find(cfg.data_type)(cfg.data)
169 system: BaseSystem = threestudio.find(cfg.system_type)(
```

基于YAML的config file

Credit: Threestudio
<https://github.com/threestudio-project/threestudio>



Q&A

在控制变量的环节，假设有 n 个布尔类型的变量，我们需要做 2^n 级别的消融实验吗？
如果只做 n 次，如何证明不是其他变量共同影响的结果。

建议：没有足够的资源去做 2^n 次实验 😊

在这些无序中找到有序，基于过去的经验对未来作出判断

控制变量的核心是指我们要run的每一个实验都有相应的对比实验来说明结果

比如：能不能先对这 n 个变量排序？哪个（或者哪个组合）最有可能有影响结果？

能不能先对 2^n 进行剪枝？删除掉一些不需要实验的组合？



课程内容

- 实验设计
 - 分解问题到最简化的实验来探索/验证技术方案
- 控制变量
 - 分析实验结果，以改进技术方案
- 可重复实验
- 实验执行
- 如何面对实验失败



可重复实验

- 为什么需要可重复实验?
 - 不仅仅是学术道德
 - 是对自己的负责
 - 如果自己的实验不可重复, 那么该如何得到正确的结论?
- 怎么做到可重复实验?
 - Coding上的注意 (e.g. random seed)
 - 代码备份, git branch/merge (代码可以追溯回到之前的历史)
 - 存log, 以及Run实验的时候复制一份代码到log



课程内容

- 实验设计
 - 分解问题到最简化的实验来探索/验证技术方案
- 控制变量
 - 分析实验结果，以改进技术方案
- 可重复实验
- 实验执行
- 如何面对实验失败

实验的执行

- 对自己为什么要跑一个实验有一个宏观的认识
 - 为什么要跑这个实验，这个实验是为了证明什么
 - 分析实验给了我们什么结果，实验A跟实验B对比可以说明结论C
 - 不要盲目得跑了一堆实验，但是没有得到有用的结果分析
 - 写一个实验列表

实验对比：

- A. Nerf + low-res diffusion model
- B. Nerf + high-res diffusion model
- C. DMTet + high-res diffusion model
- D. (DMTet + low-res diffusion model)

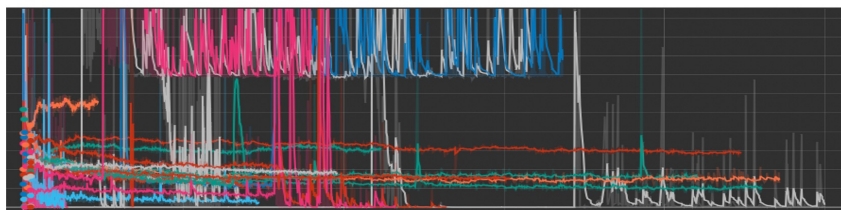


实验的执行

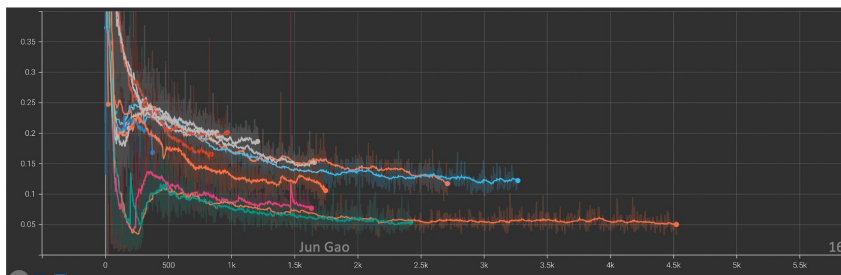
- 对自己为什么要跑一个实验有一个宏观的认识
 - 为什么要跑这个实验，这个实验是为了证明什么
 - 分析实验给了我们什么结果，实验A跟实验B对比可以说明结论C
 - 不要盲目得跑了一堆实验，但是没有得到有用的结果分析
 - 写一个实验列表

Looking at training figure

Last week



This week



Things to improve

- Geometry Side
 - Regularization loss on the geometry & SDF regularization
 - Trial with random (or sphere) initialization
 - Symmetry?
 - Network architectures
- Texture Side
 - Separate background with foreground (they should be different)
 - Other Network architecture? (2DCNN? -> needs more thinking maybe borrow CIPS-3D, check it!)
- GAN side:
 - Progressive training (both image & grid) **(done)**
 - PatchGAN
 - Noise Injection for better capacity
 - Debug the camera aware
 - Setup FID evaluation **(done)**
- Other stuffs?
 - VAE idea
 - Finetune the camera distribution for poses **(todo)**



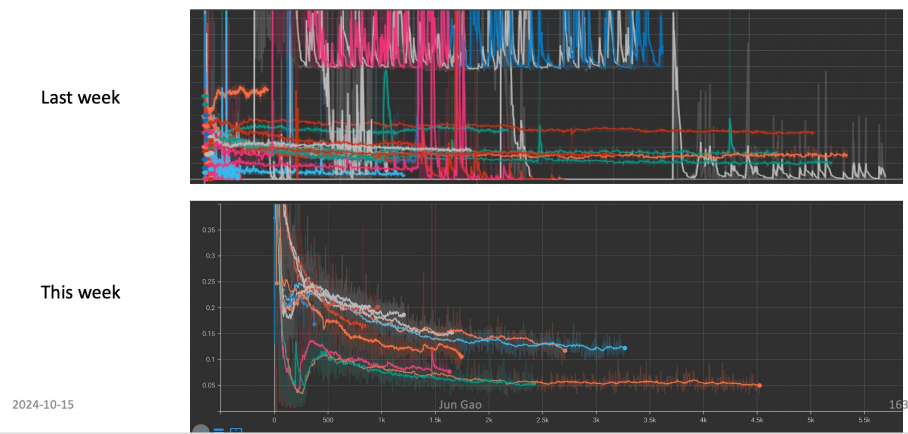
实验的执行

- 对自己为什么要跑一个实验有一个宏观的认识
- 对于实验有优先级的排序
 - 不仅是资源的限制
 - 找到idea的核心点在哪
 - 什么实验是最有可能有用的，能给我们正确的结果或者是效果提升
 - 比如DMTet + low-res diffusion prior大概率不会有用
 - 找到之前实验的问题在哪
 - 如果效果比别人差，那么具体差在什么地方？

实验的执行

- 对自己为什么要跑一个实验有一个宏观的认识
- 对于实验有优先级的排序
- 建立一个checkpoint
 - 比如每周写一个实验总结（比如PPT的形式），可以用于跟导师meet
 - 整理自己的实验思路，把它写下来（确保自己在一条正确的道路上）

Looking at training figure



Things to improve

- Geometry Side
 - Regularization loss on the geometry & SDF regularization
 - Trial with random (or sphere) initialization
 - Symmetry?
 - Network architectures
- Texture Side
 - Separate background with foreground (they should be different)
 - Other Network architecture? (2DCNN? -> needs more thinking maybe borrow CIPS-3D, check it!)
- GAN side:
 - Progressive training (both image & grid) **(done)**
 - PatchGAN
 - Noise Injection for better capacity
 - Debug the camera aware
 - Setup FID evaluation **(done)**
- Other stuffs?
 - VAE idea
 - Finetune the camera distribution for poses **(todo)**



Q&A

- 在计算资源有限的情况下，如何有效的缩短实验时间
 - 写好代码☺
 - Profile model的compute bottleneck在哪
 - GPU利用率是不是100%，如果不是，问题在哪？
 - 多卡并行会不会变慢？
 - 站在巨人的肩膀上
 - 有没有类似的代码可以借鉴？（Diffuser, NerfStudio, ThreeStudio）
 - 同一个实验室可以维护同一套代码

如何面对实验失败?

- 心态上:
 - 实验失败是科研的常态 (>90% are failure, from Kaiming He).
 - 这与个人能力没有关系, 有时候确实是idea不work。
 - 实验失败是未来成功的基石。
- 重要的是分析失败的原因: 为什么它失败了?
 - 多看一些实验的中间结果找到失败的原因
 - 比如做3D reconstruction, 需要visualize具体的3D geometry, appearance, camera 等。
 - 通过控制变量的对比实验分析
- 不要过早得宣布失败
 - 能不能排除是coding的问题?
 - 也许只要改某一个地方就成功了



Q&A

- Q: 怎样才可以宣称实验失败, idea不可行? (希望更多从个人经验感受而非方法论角度回答)
 - 举例: real-world single image 3D reconstruction
 - 为什么我觉得想法不可行? 自己的逻辑分析
- Q: 面对一个看上去失败了的实验, 如何分析其失败原因? 或者如何确认失败是由于思路问题而非实现问题?
 - 多做visualization (下节课讲)
 - 比如3D最容易弄错相机坐标, 看看RGB值域范围等
 - 把每一步的结果输出出来看看来判断他是不是Coding的问题
 - 如何分析效果不好?
 - 什么地方效果不好? 什么例子效果不好? Worst case长什么样子?



Q&A

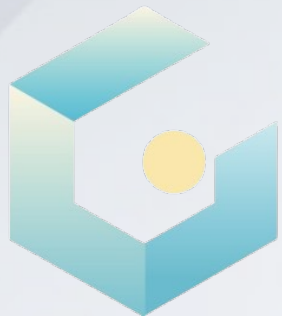
Q: 如果自己的某个idea在直觉上是非常合理、novel, 大概率有效的。但是通过实验去验证发现就是不work或者提升效果非常trivial, 那么是否还要坚持下去?

- 建议: 核心是找到它为什么不work的原因。
- 是idea本身的问题, 还是这个实验setting不适合这个idea?
- 如果确实是idea有缺陷, 有没有办法弥补?
- 如果是实验setting不对, 可不可以换个setting?
- 当这些都没有方向的时候才考虑是不是要坚持



Q&A

- Q: 在大模型时代，需要很多显卡才能跑实验。想问一下，学算法、跑实验，个人该如何选择（怕没有实验的硬件环境）
 - 建议：思考自己需要学什么样的算法，跑什么样的实验？
 - 比如：如果为了了解大模型，或者用大模型，可以只跑inference（e.g. Stable Diffusion, Stable Video Diffusion可以单卡跑inference）
 - 比如：如果为了finetune大模型，可能需要8卡甚至更多
 - 比如：如果为了训练大模型，需要更多的资源



谢谢



Sida Peng



Jun Gao



Songyou Peng



Qianqian Wang