

现代 AI Overview

Kaisen Yang

现代 AI 的四个层次

- **数学公式**：定义模型要学习的数学对象。
- **实现数学公式**：模型架构把数学对象变成可训练的计算结构。
- **实现模型的优化**：Data 决定参数更新使用什么经验。
- **研究者的优化**：Benchmark 决定方案比较与下一轮研究选择。

前两层定义并实现模型，后两层分别优化模型与研究方向。

判别式模型学习从输入到任务输出的映射

$$f_{\theta} : \mathcal{X} \rightarrow \mathcal{Y}, \quad \hat{y} = f_{\theta}(x)$$

- 分类：图像 $\rightarrow$ 类别分布。
- 回归：样本特征 $\rightarrow$ 房价、温度或坐标。
- 单目深度估计：单张 RGB 图像 $\rightarrow$ 每个像素的深度。
- 结构化预测：图像 $\rightarrow$ 分割图、检测框或关键点。
- **policy / value**：状态 $\rightarrow$ 动作分布或长期回报。

输出本身可以是一个分布；整体仍是从输入到任务输出的映射。

生成模型学习分布，也可以完成判别任务

$$p_{\theta}(x) \quad \text{或} \quad p_{\theta}(x, y), \quad p_{\theta}(y \mid x) = \frac{p_{\theta}(x, y)}{p_{\theta}(x)}$$

- 联合生成模型可以通过条件推断得到类别、深度或其他任务输出。
- AR、扩散等生成机制也可以表示分布值映射 $f_{\theta}(x) = p_{\theta}(y \mid x)$ 。
- 例：给定 RGB 图像生成深度图，均值、最大后验或采样结果都可作为预测。

常见生成建模分为六类：自回归、扩散、VAE、GAN、EBM、Normalizing Flow。

状态空间 × 生成路径形成四种建模方式

方法	状态空间	直接学习的对象	生成过程	例子
离散 AR	token、code	类别分布 $p_{\theta}(x_i \mid x_{<i})$	按位置逐个采样	语言、代码
连续 AR	连续向量、轨迹	条件密度 $p_{\theta}(x_i \mid x_{<i})$	按时间逐步采样	waveform、关节轨迹
离散扩散	token、类别	反向转移 $p_{\theta}(x_{t-1} \mid x_t)$	从 mask / 替换状态多轮恢复	掩码语言生成
连续扩散	pixel、latent	噪声、score 或速度 $v_{\theta}(x_t, t)$	反向去噪或积分 ODE	图像、音频

Flow Matching 放在连续扩散一侧：学习时间相关速度场，并沿 ODE 轨迹生成样本。

VAE、GAN 与 EBM 用三种方式表示分布

$$\text{VAE: } p_{\theta}(x) = \int p(z)p_{\theta}(x | z) \mathrm{d}z$$

$$\text{GAN: } z \sim p(z), \quad x = G_{\theta}(z), \quad p_G = (G_{\theta})_{\#}p(z)$$

$$\text{EBM: } p_{\theta}(x) = \frac{e^{-E_{\theta}(x)}}{Z_{\theta}}$$

- **VAE**: 对潜变量边缘化; 例: 从高斯 latent 解码手写数字。
- **GAN**: 生成器诱导隐式分布; 例: 一次前向生成一张人脸。
- **EBM**: 低能量表示高概率; 例: 为图像赋能量, 再从低能量区域迭代采样。

Normalizing Flow 用可逆变换定义显式密度

$$z \sim p(z), \quad x = g_\theta(z)$$

$$p_\theta(x) = p(z) \left| \det \frac{\partial g_\theta^{-1}(x)}{\partial x} \right|$$

- g_θ 必须可逆, 并且 Jacobian 行列式能够高效计算。
- **训练**: 把图像 x 逆变换为 $z = g_\theta^{-1}(x)$, 直接最大化精确对数似然。
- **生成**: 采样 $z \sim \mathcal{N}(0, I)$, 再通过 $x = g_\theta(z)$ 一次生成图像。
- **图像实例**: Glow 使用可逆 1×1 卷积和 affine coupling 逐层变换图像分布。

Normalizing Flow 学习可逆映射; Flow Matching 学习随时间变化的速度场。

不同建模方式对应不同的训练与推理过程

- **AR**: 在真实前缀上训练条件似然, 在模型前缀上逐步采样。
- **Diffusion / Flow Matching**: 在随机时刻学习局部去噪或速度, 推理时运行完整反向轨迹或 ODE。
- **VAE**: 用 ELBO 训练 encoder 与 decoder, 生成时从先验采样再解码。
- **GAN / EBM / Normalizing Flow**: 分别使用对抗信号、能量目标和精确似然; 推理算法也各不相同。
- **训推一致**: 训练过程需要覆盖推理时实际访问的状态。
- **架构适配**: 预训练、后训练与优化算法需要适配具体架构和生成接口。

架构规定信息怎样流动，也规定计算成本

- **表示**：输入进入什么状态。
- **交互**：信息从哪里聚合。
- **保留**：历史与细节如何留下。
- **输出**：得到何种局部对象。

CNN、GNN、SSM 写入较明确的结构假设；Transformer 让更多关系从 Data 中学习。

现代架构减少人工结构假设，同时保持可训练、可扩展。

四类主干写入了不同的结构假设

- **CNN**: 局部连接、参数共享与多尺度；常用于图像和空间信号。
- **RNN / SSM**: 递归状态与顺序更新；适合流式处理和长序列。
- **Transformer**: 基于内容的动态交互；用于文本、视觉和多模态。
- **GNN**: 沿显式节点与边传播；用于分子和关系结构。

MLP、残差连接与归一化，是多类主干共享的内部组件。

同一个 Transformer 可以实现多种生成任务

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right) V$$

- **因果 mask**: 前缀 $\rightarrow$ 下一 token 分布, 对应离散 AR。
- **双向可见 + 时间**: 被破坏序列 $\rightarrow$ 多位置分布, 对应掩码扩散。
- **带噪状态 + 时间**: 连续张量 $\rightarrow$ 噪声或速度, 对应连续扩散。

Transformer 定义信息交互; mask、状态、目标和输出头定义任务接口。

U-Net 与 DiT 可以实现同一个连续去噪对象

- **U-Net**: 卷积、多尺度编码-解码和跳跃连接, 保留局部细节与不同分辨率信息。
- **DiT**: 图像 token、全局 Attention 和时间条件, 沿 Transformer 路径扩展计算与容量。
- **MoE**: 在 Dense FFN 与稀疏专家之间改变容量分配。
- **量化**: 在 FP16 / BF16 与 INT8 / INT4 之间改变数值表示。

MoE 与量化可以嵌入主干, 不改变 AR 或 diffusion 的数学对象。

数学建模与模型架构是两条独立坐标

建模对象	CNN / U-Net	Transformer / DiT	RNN / SSM	GNN
映射	图像分类	文本 / 视觉分类	序列预测	节点 / 图预测
AR	PixelCNN	GPT 类模型	RNN-LM / Mamba-LM	图序列生成
diffusion / flow	U-Net 去噪	DiT 去噪	序列扩散	图扩散
policy / value	视觉策略	序列决策	流式控制	关系决策

MoE 分配容量，量化改变数值表示；两者都可以出现在矩阵中的多种组合里。

训练发生在 Data 上，梯度只能来自模型见过的经验

$$\nabla_{\theta} \hat{\mathcal{L}}_D = \sum_{i=1}^n \alpha_i \nabla_{\theta} \ell(f_{\theta}(x_i), y_i)$$

- x_i 决定模型看见什么。
- y_i 决定什么被视为正确。
- α_i 决定哪些经验被反复强化。

目标使用分布 $\rightarrow$ Data 系统 $\rightarrow$ 参数更新 $\rightarrow$ 失败回流 $\rightarrow$ Data 系统

不同数据阶段塑造不同的能力接口

- **预训练**：学习广泛表示与统计规律。
- **监督后训练**：学习任务格式与用户接口。
- **偏好 / 反馈**：塑造行为取向与拒绝边界。
- **合成数据**：控制覆盖、难度与稀缺标签。
- **交互数据**：记录行动后果与错误恢复。
- **部署回流**：补充新知识与真实失败样本。

Data 的边界，就是模型经验的边界

- **覆盖**：目标域、长尾与失败场景。
- **质量**：内容、标签与元数据。
- **配比**：每类经验在梯度中的权重。
- **污染**：训练与测试的字面或实体重叠。
- **许可**：版权、隐私与使用目的。
- **可追溯**：来源、版本与处理链。

数据规模、相关性、质量与采样权重需要一起报告。

Benchmark 通过研究者的选择间接塑造 AI

评测设计 → 分数与排名 → 发表与采用 → 资源配置 → 下一轮研究

- 能够被稳定测量的进步，更容易得到后续优化资源。
- Benchmark 影响问题选择、模型设计、数据建设和算力投入。

一个可比较的 Benchmark 必须固定五件事

$$\mathcal{B} = (\mathcal{T}, \mathcal{D}_{\text{test}}, \mathcal{P}, \mathcal{M}, \mathcal{U})$$

- 任务：做什么。
- 测试数据：在哪里做。
- 协议：工具、提示与预算。
- 指标：如何计分。
- 不确定性：方差、区间与版本。

MMLU / MMMU、SWE-bench 和 OSWorld 分别代表静态题目、代码执行与交互任务。

Benchmark 也会失效， 因此分数需要上下文

- **失效路径**

- 题库饱和， 区分窗口缩小。
- 训练污染与测试标签错误。
- judge 偏差与 Goodhart 效应。

- **阅读顺序**

- 日期、版本与训练截止时间。
- 提示、工具、重试与成本预算。
- 区间、失败案例与真实任务证据。

分数表达的是特定任务、协议与指标下的证据。

图像生成 (1/2)：数学建模与模型架构

- **数学建模**

- pixel: Pixel AR 或连续扩散。
- continuous latent: VAE codec + diffusion / Flow Matching。
- discrete code: VQ tokenizer + code AR / masked diffusion。
- 其他分布表示: Normalizing Flow 或 GAN。

- **模型架构**

- Pixel AR: 已有 pixels $\rightarrow$ PixelCNN 的 masked convolution / Transformer 的 causal attention $\rightarrow$ 下一 pixel 分布。
- latent diffusion / Flow Matching: 图像 $\rightarrow$ VAE encoder $\rightarrow z$; U-Net / DiT 在 z 上预测噪声、score 或速度; VAE decoder $\rightarrow$ 图像。
- discrete code AR: 图像 $\rightarrow$ VQ tokenizer $\rightarrow$ code grid; 因果 Transformer $\rightarrow$ 下一 code; VQ decoder $\rightarrow$ 图像。
- masked code diffusion: 带 mask 的 code grid $\rightarrow$ 双向 Transformer 多轮恢复 $\rightarrow$ VQ decoder $\rightarrow$ 图像。
- Normalizing Flow: $z \rightarrow g_\theta \rightarrow x$; GAN: $z \rightarrow G_\theta \rightarrow x$, D_ϕ 只在训练时提供信号。

图像生成 (2/2) : Data 与 Benchmark

- **Data**

- 图像、caption、编辑对与偏好反馈决定模型获得的视觉经验。
- 分辨率、裁切、过滤、去重和数据许可改变训练分布。
- codec、文本编码器和 captioner 的版本属于数据管线的一部分。

- **Benchmark**

- 视觉质量：FID、precision / recall 与人工偏好。
- 条件一致性：文本对齐、对象关系和文字渲染。
- 可控性：编辑保持、局部修改与多样性。
- 系统成本：采样步数、延迟、显存和分辨率。

单一质量分数无法同时说明对齐、覆盖、可控性与成本。

大语言模型 (1/2)：数学建模与模型架构

- **数学建模**

- 离散 AR：前缀 $\rightarrow$ 下一 token 分布。
- 离散扩散：被破坏序列 $\rightarrow$ 原 token 分布。
- 连续扩散 / Flow Matching：带噪 embedding 或 latent $\rightarrow$ 连续状态。

- **模型架构**

- 离散 AR：因果 Transformer，也可使用 RNN / SSM。
- 离散扩散：双向 Transformer + 时间或噪声条件。
- 连续生成：连续去噪器 + decoder 或 rounding。
- MoE 分配容量，量化改变部署时的数值表示。

大语言模型 (2/2)：Data 与 Benchmark

- **Data**

- 预训练：网页、书籍、论文、代码与多语言语料。
- 后训练：指令、工具轨迹、偏好、安全与失败恢复数据。
- tokenizer、采样配比、去重、许可和时间截止需要版本化。

- **Benchmark**

- 知识与推理：固定题目和封闭协议。
- 代码与 Agent：可执行测试、环境交互和终态验证。
- 行为质量：指令遵循、事实性、安全性和人工偏好。
- 系统成本：首 token 延迟、吞吐、上下文长度与生成步数。

模型分数需要与数据截止时间、工具权限和推理预算一起解释。

用四个层次分析现代 AI

- 数学公式定义模型学习的对象。
- 模型架构实现数学公式。
- Data 实现模型的优化。
- Benchmark 引导研究者的优化。

从数学对象到计算结构，从参数更新到研究选择。