MODERN AI OVERVIEW

现代 AI Overview

数学建模

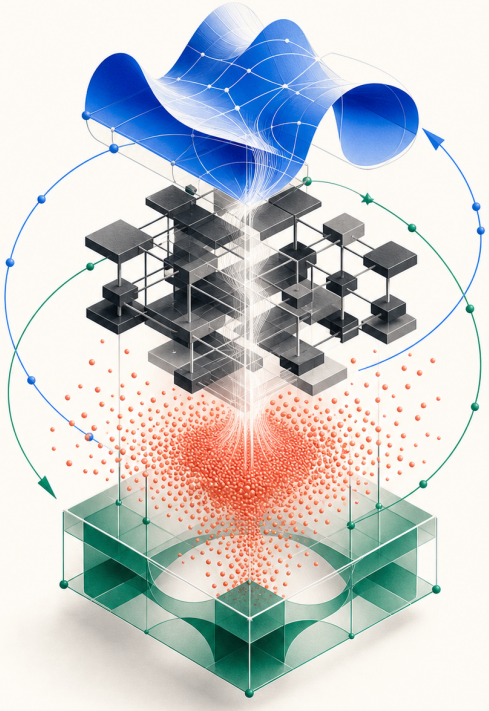
定义学习对象。

模型架构

定义计算结构。

Dataset & Benchmark

分别形成训练反馈与研究反馈。



Course AI Overview

Presented By Kaisen Yang

Date 2026 年 7 月

目录

1 Overview

本讲义提供一组分析现代 AI 系统的基本概念。面对一个模型、系统或研究工作，可以依次检查四项内容：待学习对象的数学定义、实现该对象的计算结构、用于更新模型参数的训练经验，以及用于评价和选择研究方案的评测协议。

1.1 范围与术语

人工智能（*artificial intelligence, AI*）研究和构建能够执行感知、推理、学习、规划、交流或行动任务的计算系统。**机器学习**（*machine learning, ML*）利用数据或交互经验调整模型参数，以改善给定目标或任务表现。**深度学习**（*deep learning, DL*）以多层可微参数化网络为主要模型族。**生成式 AI**（*generative AI*）学习数据分布或条件分布，并据此生成文本、图像、音频、视频、代码或动作序列。

现代 AI 系统可以组合神经网络、搜索、规则、外部工具和控制逻辑。上述术语描述不同的方法范围与系统接口；具体系统可以同时具有多种属性。深度学习的现代发展与表示学习、计算规模和数据规模密切相关 (LeCun et al., 2015)。

1.2 四个分析层次

本讲义按四个层次组织内容。每一层对应一类可明确检查的设计变量。

表 1 现代学习系统的四个层次

层次	定义内容	典型对象
数学建模	根据数据形式定义变量、条件关系与学习过程	回归/分类、AR、VAE、GAN/EBM、掩码扩散、diffusion/score、flow、policy/value/world model
模型架构	规定信息表示、交互、保留与输出，并控制计算规模	主干：CNN、RNN/SSM、Transformer、GNN；扩散网络：U-Net、DiT；容量机制：MoE；数值机制：量化
Data	构成训练过程的中心，通过 Dataset 等组织形式定义经验内容、采样分布与反馈信号	预训练语料、图文对、标签、偏好、自博弈轨迹、合成数据
Benchmark	定义评价任务、协议、指标与不确定性，并形成研究选择信号	测试集、交互环境、提示协议、指标、统计区间、成本约束

训练过程构成模型的内循环，其中 Data 是中心变量。样本内容、采样比例、标签、偏好和交互反馈决定每一步参数更新使用什么经验，也决定知识、行为模式与偏差如何进入模型。建模、架构和优化方法需要围绕目标数据分布设计，并通过数据回流持续修正。

研究开发构成外循环，其中 Benchmark 提供研究选择信号。研究者依据 Benchmark 比较方案，

继而调整问题选择、模型设计、数据建设、训练方法与计算资源。当论文发表、模型选择和资源配置依赖评测结果时，Benchmark 通过这些研究决策间接影响 AI 的发展方向。

四层之间存在明确依赖关系。数学建模确定训练目标所对应的对象；架构给出实现该对象的参数化函数族；Dataset 决定目标函数在哪些经验上计算；Benchmark 规定最终结论在什么条件下成立。后续章节将依次展开这四层，并用语言生成和图像生成两个系统说明同一数据形式下的多种建模与架构组合。

2 数学建模：判别式模型与生成模型

数学建模首先规定模型要近似什么对象。本章把常见问题分成两类。判别式模型学习从输入到任务输出的映射；生成模型学习数据本身的概率分布。两类模型都由参数 θ 表示，也都需要 Dataset 估计参数，但它们回答的问题不同。

表 2　判别式模型与生成模型的建模对象

类型	数学对象	给定的信息	模型输出
判别式模型	$f_\theta : \mathcal{X} \to \mathcal{Y}$	输入 x	$f_\theta(x) \in \mathcal{Y}$
生成模型	$p_\theta(x)$ 或 $p_\theta(x \mid c)$	无条件，或条件 c	服从数据分布的样本 x

判别式模型的输出空间 $\mathcal{Y}$ 可以是实数、类别、结构化对象，也可以是一个分布空间。当任务要求输出类别分布时，只需令 $\mathcal{Y} = \Delta(\mathcal{C})$；这仍然是映射 f_θ 的一种输出，不需要另列一个建模对象。生成模型则把完整数据对象 x 视为随机变量，直接描述它在数据空间中的分布。

2.1 判别式模型：建模输入到输出的映射

固定映射及其输出空间

给定输入空间 $\mathcal{X}$ 和输出空间 $\mathcal{Y}$，判别式模型学习映射

$$\hat{y} = f_\theta(x), \qquad f_\theta : \mathcal{X} \to \mathcal{Y}, \tag{1}$$

当输出对象本身是类别分布时，映射的陪域取为概率单纯形 $\Delta(\mathcal{C})$：

$$f_\theta(x) \in \Delta(\mathcal{C}), \qquad \hat{y} = \arg\max_{k \in \mathcal{C}} [f_\theta(x)]_k. \tag{2}$$

所谓“固定映射”，是指训练结束并固定参数 θ 后，模型对输入使用同一套映射规则。输出可以是单个数值、类别、结构化结果或一个分布；这些情况都由输出空间 $\mathcal{Y}$ 统一表示。判别式模型不需要同时给出输入的边缘分布 $p(x)$。

回归把 x 映射为连续值，分类把 x 映射为类别或类别分布，检测、分割和序列标注则把 x 映射为结构化输出。选择哪一种形式取决于数据中哪些变量被观察为输入，哪些变量被规定为任务输出。

用有限样本估计映射

给定成对数据 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$，常见估计方法是经验风险最小化

$$\hat{\theta} = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \ell\big(f_\theta(x_i), y_i\big) + \lambda \Omega(\theta). \tag{3}$$

损失 ℓ 规定预测和目标之间的误差，正则项 Ω 限制可接受的参数。若 $f_\theta(x)$ 输出类别分布，则可以最大化目标类别的预测概率：

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log[f_\theta(x_i)]_{y_i}. \tag{4}$$

分类交叉熵就是相应的负对数似然。Dataset 决定映射在哪些输入和输出上被估计，因此训练得到的映射只在数据覆盖的范围内有经验依据。

表征学习（*representation learning*）可以把映射写成 $z = e_\theta(x)$ 和 $\hat{y} = h_\phi(z)$。编码器 e_θ 保留哪些信息，取决于重建、掩码预测、对比学习、多模态对齐或监督损失等训练目标；最终仍由任务输出决定哪些表示是有用的。

策略网络与价值网络也是映射

在序贯决策中，输入变成状态或观测，输出变成动作或长期回报。策略网络学习

$$\pi_\theta : \mathcal{S} \to \Delta(\mathcal{A}), \qquad \pi_\theta(s) \in \Delta(\mathcal{A}), \qquad a \sim \pi_\theta(s), \tag{5}$$

即从状态 s 到动作分布的映射。它通过最大化轨迹上的折扣回报训练：

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right]. \tag{6}$$

价值网络学习 $V_\phi : \mathcal{S} \to \mathbb{R}$ 或 $Q_\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$，分别把状态或状态-动作对映射为期望累计回报。优势函数比较一个动作与该状态下平均水平的差异：

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s). \tag{7}$$

策略网络输出动作，价值网络输出标量评价。Actor-critic 中，actor 是策略映射，critic 是价值映射；critic 为 actor 提供学习信号，通常不直接执行动作 (Sutton and Barto, 2018)。

2.2 生成模型：建模数据分布

设观测数据来自未知分布 $p_{\text{data}}(x)$。生成模型构造参数化分布 $p_\theta(x)$，使其接近 $p_{\text{data}}(x)$：

$$p_\theta(x) \approx p_{\text{data}}(x). \tag{8}$$

无条件生成从 $p_\theta(x)$ 采样；条件生成学习 $p_\theta(x \mid c)$，其中 c 可以是文本提示、类别、另一种模态或部分观测。生成方法的差别主要在于 $p_\theta(x)$ 如何表示，以及网络直接学习哪个局部对象。

生成模型的主要类型

下面按概率分布的表示方式分类。自回归和扩散都可以处理离散变量或连续变量；表中每一行对应一种概率表示，不按应用模态重复分类。

表 3　生成模型的主要概率表示

类型	数据或状态	$p_\theta(x)$ 的表示方式
自回归	离散或连续序列	用链式法则分解为一系列局部条件分布
扩散	连续或离散状态	定义已知前向破坏过程，并学习反向生成过程
VAE	数据 x 与潜变量 z	对潜变量生成分布 $p_\theta(x \mid z)$ 做边缘化
GAN	噪声 z 与样本 x	用生成器把基础分布推送为隐式数据分布
EBM	数据 x	用能量函数定义未归一化密度
Normalizing flow	连续 z 与 x	用可逆变换和变量替换公式得到显式密度
Flow matching	连续路径 x_t	用 ODE 速度场把基础分布输运到数据分布
World model	状态、动作与后继状态	建模环境的条件转移分布

自回归：按顺序分解联合分布

建模对象：有序变量 $x = (x_1, \dots, x_T)$ 的联合分布。**自回归**（*autoregressive, AR*）使用概率链式法则

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t}), \qquad \log p_\theta(x) = \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}). \tag{9}$$

直接学习：每个位置的局部条件分布 $p_\theta(x_t \mid x_{<t})$。训练使用逐位置负对数似然；生成时按照同一顺序逐步采样。

离散自回归。 若 x_t 是词表 $\mathcal{V}$ 中的 token，局部对象是类别分布

$$p_\theta(x_t = k \mid x_{<t}) = \mathrm{softmax}(h_\theta(x_{<t}))_k, \qquad k \in \mathcal{V}. \tag{10}$$

例子：文本语言模型把一句话表示为离散 token 序列。给定前缀“今天天气”，模型输出下一个 token 的词表分布，采样得到“很好”等 token，再把新 token 加入前缀继续生成。代码和离散图像 token 使用同一建模方式。

连续自回归。 若 $x_t \in \mathbb{R}^d$，局部对象是连续条件密度。例如可以参数化为

$$p_\theta(x_t \mid x_{<t}) = \mathcal{N}(x_t; \mu_\theta(x_{<t}), \Sigma_\theta(x_{<t}))\,. \tag{11}$$

例子：机器人关节轨迹中的 x_t 是一组连续角度。模型根据此前轨迹输出下一时刻角度的均值与协方差，再从高斯密度采样，逐步生成完整运动轨迹。离散自回归与连续自回归使用相同的链式分解，区别只在于局部对象是概率质量还是概率密度。

扩散：学习破坏过程的反向过程

扩散模型先规定一个不需要学习的前向过程 $q(x_t \mid x_{t-1})$，逐步破坏数据；生成分布由学习到的反向过程 $p_\theta(x_{t-1} \mid x_t)$ 定义。连续扩散和离散扩散的状态空间不同，但都采用这一结构。

连续扩散。 **建模对象：**连续数据 $x_0 \in \mathbb{R}^d$ 的分布。高斯前向过程可以写成

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{12}$$

反向链给出生成分布

$$p_\theta(x_0) = \int p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)\, dx_{1:T}. \tag{13}$$

直接学习：反向转移，或等价参数化的噪声、干净样本与 score。例如噪声预测目标为

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|_2^2\right]. \tag{14}$$

例子：图像 latent diffusion 把一张图像编码为连续张量 $x_0 \in \mathbb{R}^{h\times w\times d}$，逐步加入高斯噪声。生成时从高斯噪声 x_T 开始，在文本条件 c 下反复预测噪声并去噪，最终得到图像 latent。DDPM 与 score-based SDE 分别给出了常用的离散时间和连续时间表述 (Ho et al., 2020; Song et al., 2021)。

离散扩散。 **建模对象：**token、类别或图结构等离散状态的分布。前向过程使用离散转移矩阵；掩码扩散是其中一种形式：

$$q(x_t^i \mid x_0^i) = \alpha_t \delta_{x_0^i} + (1 - \alpha_t)\delta_{[\texttt{MASK}]}. \tag{15}$$

离散反向链给出

$$p_\theta(x_0) = \sum_{x_{1:T}} p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t). \tag{16}$$

直接学习：反向离散转移，或被破坏位置的干净 token 分布 $p_\theta(x_0^i \mid x_t, t)$。**例子：**掩码语言扩散从全 [MASK] 序列开始。每一轮为多个位置预测词表分布，并根据采样或置信度恢复一部分 token，直到得到完整句子。D3PM 给出一般离散转移形式 (Austin et al., 2021)；MDLM 与 LLaDA 使用掩码破坏和反向 token 预测 (Sahoo et al., 2024; Nie et al., 2025)。

VAE：对潜变量边缘化

建模对象：带潜变量 z 的数据分布

$$p_\theta(x) = \int p(z) p_\theta(x \mid z)\, dz. \tag{17}$$

直接学习：生成分布 $p_\theta(x \mid z)$，以及用于近似后验的编码器 $q_\phi(z \mid x)$。训练最大化包含重建项和先验正则的变分下界 (Kingma and Welling, 2014)。**例子：**手写数字 VAE 把图像编码为低维连续向量 z；从标准高斯采样 z 后，解码器给出像素分布并生成新的数字图像。

GAN：生成器诱导的隐式分布

建模对象：基础随机变量经过生成器后诱导的分布

$$z \sim p(z), \qquad x = G_\theta(z), \qquad p_G = (G_\theta)_{\#}p(z). \tag{18}$$

直接学习：生成器 G_θ；判别器 D_ϕ 提供区分真实样本与生成样本的训练信号 (Goodfellow et al., 2014)。**例子：**人脸 GAN 把高斯向量 z 映射为人脸图像。生成时只需一次生成器前向计算，但通常不能直接计算图像的 $\log p_G(x)$。

EBM：用能量定义未归一化密度

建模对象：

$$p_\theta(x) = \frac{\exp[-E_\theta(x)]}{Z_\theta}, \qquad Z_\theta = \int \exp[-E_\theta(x)]\, dx. \tag{19}$$

直接学习：标量能量 $E_\theta(x)$，使真实数据区域的能量较低，其他区域的能量较高 (LeCun et al., 2006)。例子：图像 EBM 为每张图像输出一个能量值；真实图像应获得较低能量，经过马尔可夫链等采样过程可以从低能量区域生成图像。高维归一化常数 Z_θ 和采样过程通常难以计算。

Normalizing Flow：可逆变量变换

建模对象：由可逆映射 $x = g_\theta(z)$ 定义的显式连续密度

$$p_\theta(x) = p(z) \left| \det \frac{\partial g_\theta^{-1}(x)}{\partial x} \right|. \tag{20}$$

直接学习：可逆变换 g_θ；结构必须允许高效计算逆映射和 Jacobian 行列式 (Rezende and Mohamed, 2015)。例子：二维密度估计中，模型把标准高斯中的点通过一系列可逆变换弯曲为月牙形数据分布，同时可以精确计算每个样本的似然。

Flow Matching：学习连续速度场

建模对象：基础分布 p_0 经过 ODE 流映射后得到的目标分布

$$\frac{dx_t}{dt} = v_\theta(x_t, t), \qquad p_1 = (\Phi_\theta)_{\#} p_0, \tag{21}$$

其中 Φ_θ 是从 $t = 0$ 到 $t = 1$ 的流映射。直接学习：时间相关的速度场 $v_\theta(x, t)$，训练时回归目标向量场 (Lipman et al., 2023)。例子：图像生成可以从高斯 latent 出发，用数值积分沿速度场移动，最终得到图像 latent。Normalizing flow 直接参数化可逆映射并计算 Jacobian；flow matching 参数化连续速度场。

World Model：环境的条件分布

建模对象：给定当前状态和动作后的环境转移

$$p_\psi(s_{t+1}, r_t \mid s_t, a_t). \tag{22}$$

直接学习：后继状态、观测或奖励的条件分布。例子：机器人执行动作 a_t 后，world model 根据当前状态 s_t 生成下一状态和奖励；规划器可以重复调用该模型比较多条未来轨迹。MuZero 学习适合规划的动态表示 (Schrittwieser et al., 2020)，一般 world model 也可以让 Agent 在内部环境中预测行动后果 (Ha and Schmidhuber, 2018)。

因此，生成模型可以按照联合分布分解、反向扩散、潜变量边缘化、隐式推送、能量归一化、可逆变换、ODE 输运和条件环境转移来区分。类别之间的差异由 $p_\theta(x)$ 的表示方式和网络直接学习的局部对象共同确定。

3　模型架构：如何实现这些数学对象

3.1　架构决定信息如何流动

模型架构规定信息如何表示、交互、保留和输出。现代架构的发展方向，是减少人工写入的归纳偏置，让模型从数据中学习依赖关系，同时保持计算过程可训练、可扩展。

归纳偏置（*inductive bias*）是架构预先写入的结构假设。例如，卷积假设相邻位置更需要交互并共享局部参数；序列状态假设历史可以压缩进有限状态；图网络假设给定的边规定主要关系；Transformer 只保留 token、位置和可见性等较少的结构约束，让交互关系更多地由数据决定。

一套架构需要回答四个问题：

表 4　模型架构规定的四类信息过程

过程	需要规定的内容	例子
信息表示	原始数据如何变成向量、token、patch、节点或状态	tokenization、patch embedding、节点特征
信息交互	哪些位置能够交换信息，交互权重是否随内容变化	局部卷积、Attention、沿图边聚合
信息保留	历史与多尺度细节如何被保存并传到后续计算	递归状态、SSM 状态、残差与跳跃连接
信息输出	内部表示如何映射到建模目标需要的局部函数	分类输出、token 分布、噪声、score、速度场

架构同时规定计算成本。连接范围影响序列长度或图规模上的复杂度；状态更新影响并行训练与流式推理；缓存影响推理显存；参数激活方式影响每个样本实际使用的计算量。因此，“更少归纳偏置”需要与可训练性和扩展性共同讨论。

3.2　常见架构结构

卷积、递归或状态更新、Attention 与图消息传递是四种并列的信息交互方式。它们分别形成 CNN、RNN/SSM、Transformer 与 GNN 等主干架构家族。

表 **5**　信息交互方式与主干架构

交互方式	主干架构	写入的结构假设	信息路径
卷积	CNN	局部性、平移共享	在固定邻域内聚合，再通过堆叠扩大感受野
递归或状态更新	RNN/SSM	有序序列、历史可压缩	按序更新状态，并由状态携带历史信息
Attention	Transformer	token、位置与可见性	根据内容动态计算可见位置之间的交互权重
图消息传递	GNN	节点、边与置换对称性	沿图中给定的边聚合邻居信息

CNN 通过局部参数共享高效利用空间结构；残差连接使深层网络更容易优化 (He et al., 2016)。RNN 逐步更新隐藏状态，LSTM 通过门控控制信息保留 (Hochreiter and Schmidhuber, 1997)；SSM 用状态空间方程组织长序列，Mamba 进一步使用输入相关的选择机制 (Gu and Dao, 2023)。GNN 把局部聚合用于由节点和边组成的数据 (Kipf and Welling, 2017)。

MLP、残差连接和归一化是多种主干共享的内部结构。MLP 在同一位置内混合通道并提供非线性变换；残差连接保留已有表示并改善深层优化；归一化控制不同层中的数值尺度。它们不单独规定局部、序列、全局或图关系，因此不与四类主干家族作为同一层级比较。

3.3 Transformer：较少的结构假设

Transformer 先把数据表示为一组 token，并为 token 加入位置或结构信息。缩放点积 Attention 写成

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}+M\right)V, \tag{23}$$

其中查询 Q 根据内容与键 K 计算交互权重，再聚合相应的值 V。矩阵 M 规定可见性：因果 mask 只允许读取前缀，双向 mask 允许同一层中的全部位置交互，局部或结构 mask 可以进一步限制连接。

Transformer block 通常由多头 Attention、逐位置 MLP、残差连接和归一化组成 (Vaswani et al., 2017)。架构预先规定 token 表示、位置、mask 和参数共享方式；具体哪些可见位置需要强交互，则由 Attention 权重从数据中学习。

与其他主干相比，CNN 预先限定局部邻域，RNN/SSM 预先规定状态随序列演化，GNN 预先给出节点之间的边。Transformer 对可见 token 之间的关系设置较少的硬编码限制，因此可以在文本、图像、音频、多模态和决策序列上复用。标准全 Attention 的计算和中间激活随 token 数量呈二次增长；减少结构假设会把一部分代价转移到数据规模和计算预算上。

3.4 扩散模型中的 U-Net 与 DiT

连续扩散需要网络近似噪声 $\epsilon_\theta(x_t, t, c)$、score、干净样本或速度场。U-Net 与 DiT 是实现这些局部函数的两类常见网络；扩散的前向破坏过程和反向生成分布保持不变。

表 6　U-Net 与 DiT 作为扩散去噪器的比较

结构	信息组织方式	主要计算特征
U-Net	下采样获得多尺度表示，上采样恢复分辨率，并通过跳跃连接保留对应尺度的细节	具有明确的空间层级和局部结构，计算量随各尺度分辨率分配
DiT	把图像 latent 划分为 token，加入时间与条件信息，再使用 Transformer block 交互	关系由 Attention 从数据学习，扩展方式接近视觉 Transformer

U-Net 使用编码-解码结构与跳跃连接处理多尺度空间信息 (Ronneberger et al., 2015)。DiT 用 Transformer 处理带噪图像 latent 或 patch，并输出扩散目标所需的连续张量 (Peebles and Xie, 2023)。二者在这里可以并列，因为它们都实现同一个扩散局部函数；差异集中在信息组织方式和计算扩展路径。

3.5 MoE：计算与容量分配

Dense FFN 对每个输入激活同一个前馈网络。**混合专家**（*mixture of experts, MoE*）设置多个专家 F_e，由路由器 $g(x)$ 为每个 token 选择少量专家：

$$y = \sum_{e \in \mathrm{TopK}(g(x))} \alpha_e(x) F_e(x). \tag{24}$$

当专家数量增加而每个 token 只激活少数专家时，总参数量可以显著大于单次前向计算使用的激活参数量 (Shazeer et al., 2017)。路由器需要学习专家选择，训练还要处理负载均衡、跨设备通信、专家容量和路由稳定性。

MoE 通常替换 Transformer 或 DiT block 中的 Dense FFN，也可以嵌入其他主干。它改变参数容量和计算分配，不规定数据是 AR、diffusion、分类还是策略学习对象。Dense FFN 与 MoE 构成同一层级的对照；MoE 与 U-Net、DiT 分属容量机制和任务网络两个层级。

3.6 模型量化：数值表示与部署

模型量化（model quantization）降低权重、激活或缓存的数值精度。它不改变层之间的连接关系，也不改变模型需要近似的数学对象；它用低比特数值近似原模型中的高精度计算。

一种常见的仿射量化把浮点权重 w 映射为整数 q，并在计算中恢复近似值 $\hat{w}$：

$$q = \mathrm{clip}\left(\mathrm{round}\left(\frac{w}{s}\right) + z_0, q_{\min}, q_{\max}\right), \qquad \hat{w} = s(q - z_0), \tag{25}$$

其中 s 是缩放因子，z_0 是零点。缩放参数可以按整个张量、通道或参数分组估计；分组越细，通常越容易适应不同区域的数值范围，同时需要保存更多缩放元数据。

量化对象和训练阶段需要分别说明：

表 7　模型量化的主要维度

维度	常见形式	含义
量化对象	权重、权重与激活、KV cache	决定哪些张量使用低比特表示
数值精度	FP8、INT8、INT4 等	决定存储范围、分辨率与硬件计算形式
参数粒度	per-tensor、per-channel、per-group	决定多少数值共享同一组缩放参数
应用阶段	PTQ、QAT	分别表示训练后量化和量化感知训练

训练后量化（*post-training quantization, PTQ*）在模型训练完成后估计量化参数；**量化感知训练**（*quantization-aware training, QAT*）在训练或微调时模拟量化误差，让参数适应低精度计算。权重量化主要降低模型存储和权重读取成本；激活与 KV-cache 量化还会影响中间状态的显存和计算。以原始权重存储估算，7B 参数模型使用 FP16 约需 14GB，使用 INT4 约需 3.5GB；实际部署还需要缩放参数、缓存和运行时内存。

量化效果取决于数值离群值、校准数据、分组方式和硬件内核。更低比特数可以减少存储与带宽，同时引入更大的近似误差。因此，量化需要同时报告精度、内存、吞吐、延迟和实际硬件环境。

3.7　数学建模与架构的组合关系

数学建模规定需要近似的对象，架构规定用于近似该对象的函数族与信息流。同一主干可以实现多种数学对象，同一数学对象也可以由不同主干实现。

表 8　架构概念中可以并列比较的层级

层级	并列对象	比较内容
信息交互方式	卷积、递归或状态更新、Attention、图消息传递	信息沿什么路径流动
主干架构家族	CNN、RNN/SSM、Transformer、GNN	如何组合交互方式与共享组件
扩散局部函数网络	U-Net、DiT	如何实现噪声、score 或速度预测
容量分配机制	Dense FFN、MoE	每个输入激活哪些参数与计算
数值表示与部署机制	FP16、FP8、INT8、INT4；PTQ、QAT	参数、激活与缓存采用什么精度

表 9　数学建模与主干架构的组合示例

建模对象	CNN 系	Transformer 系	RNN/SSM	GNN
任务映射 f_θ	图像分类与分割	文本、视觉与多模态预测	序列预测	节点与图预测
AR 分布	PixelCNN	离散 token 或连续轨迹 AR	RNN-LM、SSM-LM	图序列生成
扩散分布	U-Net 连续去噪	DiT 连续去噪、Transformer 掩码扩散	序列扩散	分子与图扩散
决策与动态模型	视觉控制与动力学	Agent、序列决策与世界模型	流式控制	关系决策与图动力学

以 Transformer 为例：双向可见性和任务输出头可以实现分类；因果 mask 与词表输出头可以实现离散自回归；带时间条件的双向 Transformer 可以实现掩码扩散；输入带噪连续 patch 并输出噪声、score 或速度时，可以实现连续扩散。Transformer 主干保持相似，输入表示、mask、输出头、损失和采样过程随数学对象改变。

MoE 不作为上表中的主干列。它可以加入 Transformer、DiT 或其他网络，在不改变数学对象的前提下扩大参数容量。量化同样适用于不同主干、任务网络和容量机制，它改变数值表示与部署成本。由此，主干架构、任务网络、容量机制和数值机制可以分别比较，再在具体系统中组合。

4 Data：训练模型的中心

本章用 Data 表示训练所使用的全部经验，包括静态样本、标签、偏好、交互轨迹、奖励、合成样本和部署回流；Dataset 表示其中经过组织、版本化和可复用的数据集合。对机器学习系统，Data 是训练过程的中心：模型参数只能根据进入训练过程的经验获得更新。

以带权经验风险为例，训练目标及其梯度为

$$\widehat{\mathcal{L}}_D(\theta) = \sum_{i=1}^{n} \alpha_i \, \ell(f_\theta(x_i), y_i), \qquad \nabla_\theta \widehat{\mathcal{L}}_D = \sum_{i=1}^{n} \alpha_i \, \nabla_\theta \ell(f_\theta(x_i), y_i). \tag{26}$$

样本 x_i 的内容、目标 y_i 的定义和采样权重 α_i 直接决定梯度由哪些经验构成。自监督学习中的目标可以从 x_i 构造，强化学习中的样本可以是带奖励的轨迹；这一关系仍然成立。架构规定模型能够表达哪些函数，Data 决定训练实际推动模型学习哪些规律。

以 Data 为中心意味着先描述目标使用分布、关键能力和失败样本，再设计数据来源、采集、标注、过滤、去重、配比、课程顺序和反馈回流。模型结构与训练算法需要根据它们利用这些数据的效果进行选择。

4.1 采样分布与任务定义

从数学上看，训练集是目标分布的有限样本；从工程上看，它是文件、字段、标签、过滤器和许可协议；从认识论上看，它决定模型能直接从哪些经验中归纳。三种视角缺一不可。

经典 ImageNet 通过 WordNet 层级、类别定义与标注流程，把“识别物体”形式化为大规模封闭类别分类问题 (Deng et al., 2009)。网页语料同时带来语言知识、网页分布、重复、时间边界和社会偏差。图文对数据常把 alt-text 或抓取文本作为弱监督，因此模型学习网络文本与图像之间的统计对应。LAION-5B 展示了这种大规模开放图文数据范式 (Schuhmann et al., 2022)。

> 直觉
>
> Dataset 是经过传感器、采样规则、标签体系和清洗流程形成的世界切片。它的覆盖范围限定了模型能够直接获得的训练经验。

4.2 一条数据记录包含哪些隐含决定

对样本 (x, y, m)，x 是观测，y 是标签或目标，m 可以包括来源、时间、语言、许可、标注者、设备等元数据。每个字段背后都有设计决定：

- **采样单位**：一张图、一段视频、一轮对话，还是完整任务轨迹？
- **标签空间**：类别是否互斥？“其他”如何处理？谁有权定义正确答案？
- **时间范围**：数据在哪个日期截断？知识更新如何进入模型？
- **群体覆盖**：语言、地区、设备、场景和用户是否代表部署环境？
- **依赖结构**：样本是否真正独立？同一网页、仓库或人物是否跨划分重复？

随机划分只有在样本近似独立同分布时才提供可靠估计。若相邻视频帧、同一病人记录或同一代码仓库同时出现在训练与测试集，模型可能利用实体记忆而非任务泛化。时间划分、实体划分和域外划分往往比简单随机划分更严格。

4.3 数据在现代模型生命周期中的不同角色

现代基础模型通常依次使用预训练、监督后训练、偏好反馈、合成和交互等多阶段数据。

表 10 不同训练阶段的数据角色

阶段	主要数据	主要作用
预训练	网页、书籍、代码、图文、音视频	学习广泛表示、统计规律和基础生成能力
监督后训练	指令-回答、任务示例、专家演示	建立可用接口和任务格式
偏好/反馈	成对偏好、排序、批注、奖励	调整帮助性、安全性、风格和拒绝行为
合成数据	模型生成、程序生成、仿真数据	扩覆盖、控难度、补稀缺标签，但可能继承模型偏差
交互数据	环境轨迹、工具调用、游戏对局	学习行动后果、错误恢复与长期信用分配
持续更新	新知识、失败样本、领域回流	修复时间边界和部署分布漂移

T5 的 C4 数据处理说明了网页语料清洗与统一 text-to-text 接口如何共同塑造训练 (Raffel et al., 2020)。InstructGPT 展示了示范、偏好比较与强化学习后训练的组合 (Ouyang et al., 2022)。Self-Instruct 则展示用模型生成指令数据扩展监督信号 (Wang et al., 2023)。

每个阶段都在改变条件分布。预训练可能让模型“知道”某种模式，后训练决定它是否会在用户接口中稳定表达；偏好优化提高可接受性，却也可能压低少见但正确的回答；交互训练获得反馈，却容易过拟合特定环境或 verifier。

4.4 数据质量的六个维度

“高质量数据”至少包含六个相互冲突的维度：

1. **正确性**：内容、标签和元数据是否可靠；
2. **覆盖度**：是否包含目标分布中的长尾和失败场景；
3. **多样性**：是否避免大量近重复样本占据梯度；
4. **可学习性**：格式、难度和噪声是否提供有效信号；

5. **合法与合规**：许可、隐私、敏感信息和使用目的是否匹配；
6. **可追溯性**：能否回答样本来自哪里、经历了什么处理。

清洗强度需要在噪声控制与覆盖保持之间权衡。过度过滤可能删除方言、少数群体或困难样本；去重可以减少记忆和污染，也可能降低真实世界中高频事实的权重。数据配比本质上是优化预算分配：一个样本被重复十次，相当于在经验风险中提高其权重。

例子：为什么“更多数据”可能更差

若给医学问答模型加入大量通用网页文本，token 数增加了，但医学证据在混合中的相对权重下降；若加入未经校验的合成答案，模型可能学会流畅的错误模式。数据规模、质量、相关性与训练配比必须一起讨论。

4.5 污染、记忆与泛化

当测试题、答案、近重复版本或解题讨论进入训练集，Benchmark 分数会混合记忆与泛化。污染并不总是故意作弊：开放网络数据与公开题库天然重叠，代码仓库的 issue、修复 commit 与测试用例也可能互相泄漏。

应分别报告：

- 字面重叠与近重复检测；
- 题目发布日期是否晚于训练截止日期；
- 实体、模板、来源网站或仓库级别的划分；
- 已知污染子集上的敏感性分析；
- 模型是否能解释、迁移或在扰动版本上保持表现。

语言知识、事实和代码模式的获得需要记忆。评测需要进一步区分已有内容的复现、组合迁移与新任务泛化，因此 Dataset 与 Benchmark 应共同设计。

常见误区

训练过程的透明度需要同时记录数据版本、过滤、去重、采样权重、tokenizer、数据顺序和后训练混合。原始数据受限时，仍可发布数据说明、统计和审计方法。

本节结论

Data 是训练模型的中心。它决定哪些经验产生梯度、哪些模式被反复强化、什么被视为正确，以及哪些情况从未进入训练过程。Dataset 建设因此是模型能力、边界与可靠性设计的核心工作。

5 Benchmark：研究反馈与能力证据

Benchmark 把复杂的模型行为压缩为可比较的结果，并通过研究者的选择影响 AI 的发展。它形成一条外部反馈链：

$$\text{Benchmark 设计} \longrightarrow \text{分数与排名} \longrightarrow \text{发表、采用与资源配置} \longrightarrow \text{下一轮研究决策}. \tag{27}$$

研究者会把时间和算力投入能够被当前评测识别的改进。任务、协议和指标因此决定哪些能力进步更容易被看见、比较和奖励。若评测强调静态题目，研究会持续优化静态题目表现；若评测同时计入鲁棒性、成本、安全和真实任务完成率，相应方向会获得更明确的优化信号。Benchmark 对 AI 发展的影响是间接的，但这种影响会在连续的模型选择和资源投入中累积。

5.1 Benchmark 的组成

一个完整 Benchmark 应写成

$$\mathcal{B} = (\mathcal{T}, \mathcal{D}_{\text{test}}, \mathcal{P}, \mathcal{M}, \mathcal{U}), \tag{28}$$

其中 $\mathcal{T}$ 是任务定义，$\mathcal{D}_{\text{test}}$ 是测试实例，$\mathcal{P}$ 是推理协议，$\mathcal{M}$ 是指标，$\mathcal{U}$ 是不确定性与报告规则。只给出测试数据而不固定协议，结果通常不可比较。

协议可能包括：zero-shot 或 few-shot、系统提示、是否允许工具、采样温度、最大 token、尝试次数、时间预算、硬件、环境版本、失败重试和人工干预。对于 Agent，还必须固定初始状态、可用动作、环境随机性和终止条件。

> **直觉**
>
> 同一组测试题在开卷、闭卷、允许工具、不同时间限制或不同重试次数下会产生含义不同的分数。因此，测试实例必须与推理协议和评分规则一起报告。

5.2 指标必须对应失败成本

分类准确率

$$\text{Acc} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[\hat{y}_i = y_i] \tag{29}$$

隐含每个样本权重相同、每类错误成本相近。类别不平衡时，宏平均 F1、ROC-AUC、PR-AUC 或每类召回更合适。概率预测还应评估校准，例如 Brier score：

$$\text{Brier} = \frac{1}{n}\sum_{i=1}^{n} \|p_i - y_i\|_2^2. \tag{30}$$

生成任务没有唯一答案，常用似然、BLEU/ROUGE、FID、CLIP 相似度、执行测试、人类偏好或 LLM judge。每个指标覆盖能力的一个投影：单元测试度量可执行正确性，可维护性需要额外审查；图文对齐与视觉伪影需要分别评估；偏好胜率与事实正确性也需分别测量。

Agent 的最终成功率比单步准确率更接近任务目标，但仍应同时报告：部分完成、平均步骤、延迟、token/金钱成本、失败类型、安全违规与跨随机种子方差。若一个方法只靠更多采样和重试提高成功率，必须把额外计算作为结果的一部分。

例子：准确率相同，风险不同

两个医疗分类器准确率都为 95%。模型 A 的错误集中在轻症，模型 B 漏掉大量危重病例。单一准确率无法表达部署风险；应根据错误成本报告灵敏度、特异度、校准和分组表现。

5.3 代表性 Benchmark 的“测量对象”

表 **11**　代表性 Benchmark 及其证据边界

Benchmark	主要任务与协议	它提供的证据	证据边界
ImageNet	封闭类别图像分类	对特定视觉类别分布的识别能力	通用视觉理解或开放世界可靠性
MMLU	57 个学科的选择题	多学科知识与选择题推断 (Hendrycks et al., 2021)	开放式研究、行动或事实校准
MMMU	多学科、多图像类型的专家题	视觉信息与领域知识联合推理 (Yue et al., 2024)	真实多模态工作流完成能力
SWE-bench	给定 issue 和代码仓库，提交补丁并运行测试	跨文件理解、编辑与执行验证 (Jimenez et al., 2024)	任意软件工程生产力或代码质量
OSWorld	在真实桌面应用中执行开放任务	GUI grounding、跨应用行动与终态验证 (Xie et al., 2024)	无约束真实电脑使用的安全可靠性
HELM	多场景、多指标统一协议	准确率之外的校准、鲁棒、公平、效率等 (Liang et al., 2022)	一个总分概括所有价值维度
HLE	专家设计的高难度多学科闭合题	前沿学术知识与可自动核验答案 (Phan et al., 2025)	开放研究能力或长期自主工作

这些 Benchmark 分别覆盖知识问答、多模态理解、代码修改和 GUI 操作等能力切片。不同任务的分数应在各自协议和证据边界内解释，跨任务能力需要额外评测。

5.4 静态题库、生成评测与交互评测

静态题库输入固定、成本低、重复性强，适合快速比较；但容易污染、饱和和被模板化优化。生成评测允许开放输出，更接近真实交流；但评分依赖参考答案、人类或 judge，方差和偏见更大。交互评测让系统影响环境，可以测工具使用、恢复与长期规划；但环境维护、状态复现与安全成本更高。

更接近真实任务通常意味着更高的测量成本和更低的可控性。完整评测体系可以按成本与真实性形成证据阶梯：

1. 单元能力与受控扰动；
2. 综合任务与执行验证；
3. 交互环境与长期任务；
4. 小规模真实用户研究；
5. 部署监控、事故与分布漂移审计。

5.5 Benchmark 的失效机制

5.5.1 饱和

当大量模型接近满分，分数对前沿差异不再敏感。HLE 在传统学术题库饱和背景下构造了更困难的闭合题 (Phan et al., 2025)。难度提升的有效期仍然有限：2026 AI Index 指出，前沿模型在 HLE 上一年内提高约 30 个百分点，许多原本希望长期有效的评测在数月内迅速压缩区分窗口 (Stanford Institute for Human-Centered Artificial Intelligence, 2026)。

5.5.2 数据与标签错误

测试集的标签与任务定义也需要审计。标签错误会改变模型排名，且高性能模型可能更多暴露测试标签问题。对多个经典数据集的系统检查显示，测试集标签错误可能显著影响 Benchmark 稳定性 (Northcutt et al., 2021)。题目审计、争议标记和版本化是评测维护的一部分。

5.5.3 污染与适配

公开题库可能进入预训练数据；提示模板、输出格式和 judge 偏好也可能被专门适配。即使没有看到具体答案，围绕 Benchmark 的大量开发也会让验证集逐渐变成训练信号。隐藏测试集、滚动更新和时间划分可以减轻问题，真实域外验证仍然必要。

5.5.4 LLM-as-a-judge 偏差

LLM judge 可以扩大开放式评测规模，但会受到位置、冗长、自我偏好和推理能力限制 (Zheng et al., 2023)。应随机化答案顺序、使用明确 rubric、报告多 judge 或人类校准，并保留原始输出供复查。

5.5.5 Goodhart 效应

当一个测量值成为优化目标，它与真实目标之间的相关性可能下降 (Goodhart, 1984)。团队不断针对排行榜调参，会找到指标漏洞、格式捷径或环境脆弱点。真正稳健的做法是多指标、隐藏测试、扰动测试、失败分析和部署反馈共同构成证据。

常见误区

“Benchmark 上超过人类”表示模型在该任务、数据、协议和指标下超过某个人类基线。对整个学科的掌握程度和开放环境可靠性需要另外设计证据。

5.6 排行榜的阅读方法

读任何表格前，按以下顺序检查：

1. 比较的是基础模型、系统还是带工具和多次采样的 Agent？
2. 模型是否使用相同提示、工具、token、时间和成本预算？
3. 测试数据与训练截止时间是否可能重叠？
4. 指标是否匹配真实失败成本，是否报告置信区间？
5. 排名差异是否大于随机性、judge 方差与环境噪声？
6. 是否公开失败案例，还是只有平均分？

本节结论

Benchmark 同时承担两项作用：它约束能力主张，也向研究者提供优化信号。一个分数需要连同任务、协议、指标和不确定性一起解释；一套评测体系还需要检查它正在把研究资源引向哪些能力。

6 两个系统案例：语言生成与图像生成

语言和图像都是应用模态。同一模态可以采用多种数学建模，同一种建模也可以由不同架构实现。为了完整描述一个系统，本章把每个案例放在同一个小节内，并依次回答五个问题：

1. **表示**：原始数据以 token、pixel、连续 latent 还是离散 code 进入模型；
2. **数学建模**：模型如何表示数据分布，网络直接学习哪个局部对象；
3. **架构**：信息如何交互、保留和输出；
4. **Database**：训练数据如何收集、组织、版本化并构造成训练实例；
5. **Benchmark**：使用什么任务、协议和指标比较系统。

这里的 Database 是数据层的工程称谓，覆盖 Data、版本化 Dataset、元数据、存储索引和训练数据管线。表示在案例中单独列出，因为 token、pixel 与 latent 的选择会直接改变后续数学对象。

6.1 案例一：语言生成系统

6.1.1 表示

文字生成系统通常先用 tokenizer 把字符串变成离散序列

$$x_0 = (x_0^1, \dots, x_0^L), \qquad x_0^i \in \mathcal{V}, \tag{31}$$

其中 $\mathcal{V}$ 是词表。离散自回归和离散扩散直接在 token 空间建模。连续扩散进一步使用 embedding 或编码器

$$z_0 = E(x_0) \in \mathbb{R}^{L\times d} \tag{32}$$

得到连续序列，生成结束后再通过解码器或 rounding 映射回 token。词表、分词粒度、序列长度和连续表示维度共同规定模型实际看到的语言对象。

6.1.2 数学建模

语言生成可以采用离散自回归、离散扩散或连续扩散。三种方法最终都输出 token 序列，但概率对象和生成路径不同。

表 12　语言生成的三种数学建模方式

方式	模型状态	网络直接学习的对象	生成路径
离散自回归	已生成的 token 前缀 $x_{<i}$	下一个 token 分布 $p_\theta(x_i \mid x_{<i}, c)$	从左到右逐 token 采样
离散扩散	被掩码或替换的离散序列 x_t	反向转移或干净 token 分布 $p_\theta(x_0^i \mid x_t, t, c)$	多轮恢复一个或多个位置
连续扩散	连续文本表示 $z_t \in \mathbb{R}^{L\times d}$	噪声、score、干净表示或速度	连续去噪后解码为 token

离散自回归。 自回归模型直接分解离散序列的联合分布：

$$p_\theta(x_0 \mid c) = \prod_{i=1}^{L} p_\theta(x_0^i \mid x_0^{<i}, c). \tag{33}$$

训练样本由“前缀-下一个 token”构成，网络在每个位置输出词表上的 softmax 分布。生成从起始 token 开始，每次采样一个 token 并追加到前缀。

具体例子：给定提示“请用一句话解释光合作用”，模型先计算第一个回答 token 的分布；采样后把它加入上下文，再计算第二个 token。整段回答的概率等于这些局部条件概率的乘积。

离散扩散。 离散扩散直接在词表状态上定义前向破坏过程。以掩码扩散为例，训练时随机选择时间 t，把一部分 token 替换为 [MASK]，网络根据仍然可见的左右文和时间条件预测原 token，具体前向过程见式 (15)。生成时可从全掩码序列开始：

$$[\mathsf{MASK}, \ldots, \mathsf{MASK}] \longrightarrow x_T \longrightarrow x_{T-1} \longrightarrow \cdots \longrightarrow x_0. \tag{34}$$

每轮可以同时更新多个位置，更新顺序可以由时间表、随机采样或 token 置信度决定。D3PM 给出一般离散转移，MDLM 和 LLaDA 展示了面向语言的掩码扩散实现 (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025)。

具体例子：生成“上海是中国最大的城市之一”时，中间状态可能是“上海 [MASK] 中国 [MASK] 城市之一”。模型同时预测两个掩码位置的词表分布，并在后续轮次重新检查低置信度位置。生成顺序由反向扩散过程决定，可以与阅读顺序不同。

连续扩散。 连续扩散先把离散文本映射为连续表示 $z_0 = E(x_0)$，再进行高斯加噪：

$$z_t = \alpha_t z_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{35}$$

去噪器学习 $\epsilon_\theta(z_t, t, c)$、score、$z_0$ 或速度。生成从连续高斯噪声开始，经过多步去噪得到 $\hat{z}_0$，最后由解码器 $D(\hat{z}_0)$ 或 rounding 机制恢复离散 token。Diffusion-LM 是在连续词向量空间进行文本扩散的代表性例子 (Li et al., 2022)。

具体例子：做带属性控制的句子生成时，可以让条件 c 表示主题或情感。去噪网络在每一步同时调整整段连续序列，最后解码出完整句子。连续状态便于施加梯度引导或连续控制，同时需要处理连续表示与离散词表之间的解码误差。

6.1.3 架构

表 13　语言数学建模与架构的对应关系

数学建模	信息可见性与输出	可用架构
离散自回归	读取前缀；输出词表分布	带因果 mask 的 decoder-only Transformer；RNN；SSM
离散扩散	读取当前被破坏序列的左右文；输出多个位置的词表分布	带时间条件的双向 Transformer
连续扩散	读取整段连续噪声状态；输出同形状的连续张量	带时间条件的双向 Transformer 去噪器；序列 U-Net

因果 mask、双向可见性和输出头随数学建模改变。MoE 可以替换 Transformer 中的 Dense FFN 来增加容量，量化可以降低权重、激活与 KV cache 的部署成本；这些机制保留原有的 AR 或 diffusion 概率对象。

6.1.4 Database

语言系统的 Database 层包括网页、书籍、代码、对话、指令、偏好和安全数据，以及来源、语言、时间、许可、去重状态等元数据。同一原始语料会按建模方式构造成不同训练实例：

- 离散 AR 使用移位后的 token 形成“前缀–下一个 token”目标；
- 离散扩散为序列采样时间、掩码比例和被破坏位置；
- 连续扩散调用文本编码器得到 z_0，再采样时间与连续噪声。

数据系统需要版本化 tokenizer、语料配比、序列长度、破坏时间分布、连续编码器和条件字段。预训练语料提供语言规律，指令与偏好数据进一步决定系统在用户接口中的行为。

6.1.5 Benchmark

语言系统可以分别评估语言质量、事实性、推理、代码执行、指令遵循、可控生成、安全和人类偏好。MMLU/HLE 提供闭合知识题证据，SWE-bench 测试仓库级代码修改，开放生成还需要人类评价或经过校准的 judge。

比较三种建模时，需要固定提示、输出长度、采样次数、温度、扩散步数、延迟和计算预算。AR 的逐 token 似然、扩散的训练界和开放生成质量对应不同证据；系统报告应同时给出任务表现和实际生成成本。

6.2 案例二：图像生成系统

6.2.1 表示

设图像为 $x \in [0,1]^{H\times W\times C}$。图像生成常用三种状态空间：

$$\underbrace{x}_{\text{pixel}} \longrightarrow \underbrace{z = E_\phi(x) \in \mathbb{R}^{h\times w\times d}}_{\text{continuous latent}} \quad \text{或} \quad \underbrace{u = Q(E_\phi(x)) \in \{1,\dots,K\}^{h\times w}}_{\text{discrete code}}. \tag{36}$$

Pixel 表示保留原始分辨率与颜色通道；连续 latent 通过 encoder 压缩图像；离散 code 再把 latent 量化为 codebook 索引。分辨率、压缩率、codebook 大小和重建误差决定后续模型接收多少视觉信息。

6.2.2 数学建模

AR、VAE、diffusion、flow 和 GAN 描述图像分布的方式不同。它们可以作用在 pixel、连续 latent 或离散 code 上。

表 14　图像生成的数学建模对象

方案	表示空间	概率对象或局部目标	生成过程
Pixel AR	pixel	$\prod_i p_\theta(x_i \mid x_{<i})$	按像素、通道或子像素顺序生成
Pixel diffusion	pixel	像素张量上的反向高斯过程	从像素噪声多步去噪
VAE	连续 latent 与 pixel	$p(z)p_\psi(x \mid z)$ 与近似后验 $q_\phi(z \mid x)$	采样 z 后一次解码
Latent diffusion/flow	连续 latent	$p_\theta(z)$ 上的噪声、score 或速度	在 latent 中去噪或积分，再解码
离散 latent 生成	离散 code	code 序列分布 $p_\theta(u)$	逐 code 生成或多轮恢复，再解码
Normalizing flow	pixel 或连续 latent	可逆映射及显式密度	从基础分布经可逆变换
GAN	pixel 或连续 latent	$x = G_\theta(z)$ 诱导的隐式分布	一次生成器映射

Pixel AR 与 Pixel diffusion。 **Pixel AR** 选定光栅或其他顺序，把整张图像分解为像素条件分布。它具有显式似然，生成需要大量串行步骤 (van den Oord et al., 2016)。

Pixel diffusion 把整张像素张量作为 x_0，直接加入高斯噪声并学习反向去噪 (Ho et al., 2020)。Pixel 级模型保留全部图像细节，训练与采样的中间张量随原始分辨率增长。

VAE 与 latent diffusion。 VAE 作为独立生成模型时定义

$$p_\psi(x) = \int p(z)p_\psi(x \mid z)\, dz, \tag{37}$$

编码器 $q_\phi(z \mid x)$ 用于训练近似后验，解码器 $p_\psi(x \mid z)$ 把 latent 还原为像素 (Kingma and Welling, 2014)。标准高斯 $p(z)$ 直接作为生成先验时，从高斯采样一次即可解码图像。

在 latent diffusion 中，VAE 主要承担压缩与解码。图像先经 $z = E_\phi(x)$ 进入低分辨率连续空间，再学习更强的 latent 先验 $p_\theta(z)$：

$$p(x) = \int \underbrace{p_\psi(x \mid z)}_{\text{VAE decoder}} \underbrace{p_\theta(z)}_{\text{diffusion or flow prior}} dz. \tag{38}$$

$p_\theta(z)$ 可以由 diffusion 的噪声/score 预测或 flow matching 的速度场定义。生成先在 latent 中完成，再由 VAE decoder 恢复像素。Latent Diffusion Models 展示了这一组合 (Rombach et al., 2022)。

离散 latent。 VQ-VAE 把图像压缩为 codebook 中的离散索引 u，解码器负责从 code 网格重建图像 (van den Oord et al., 2017)。随后可以对 u 使用离散 AR 逐 code 生成，也可以使用掩码扩散多轮恢复 code。这里的数学对象位于离散 code 空间，VQ-VAE 提供图像与 code 之间的接口。

Normalizing flow 与 GAN。 Normalizing flow 用可逆映射 $x = g_\theta(z)$ 和变量替换公式计算显式密度。精确似然要求变换能够高效求逆并计算 Jacobian 行列式；Glow 是这种建模方式的代表 (Kingma and Dhariwal, 2018)。

GAN 用生成器 G_θ 把基础噪声直接映射为图像，并由判别器提供训练信号 (Goodfellow et al., 2014)。它支持一次前向生成，通常不提供可直接计算的图像似然。

6.2.3 架构

表 15　图像数学建模与架构的对应关系

数学建模	核心架构	架构承担的工作
Pixel AR	PixelCNN 的 masked convolution；因果图像 Transformer	保证当前位置只能读取此前像素或 patch，并输出局部条件分布
Pixel diffusion	多尺度 U-Net；pixel patch DiT	输入带噪像素与时间，输出噪声、score 或速度
VAE	CNN/ResNet encoder 与 decoder	编码后验参数并从 latent 重建图像
Latent diffusion/flow	VAE codec 加 U-Net 或 DiT	codec 转换 pixel/latent；主干预测 latent 中的噪声、score 或速度
离散 latent 生成	VQ-VAE tokenizer 加因果或双向 Transformer	tokenizer 转换图像/code；Transformer 逐 code 生成或多轮恢复
Normalizing flow	coupling layer、可逆 1×1 convolution	保持变换可逆并高效计算 Jacobian
GAN	CNN 或 Transformer 生成器与判别器	生成器产生样本，判别器提供对抗训练信号

6.2.4 Database

图像系统的 Database 层保存原始图像、图文对、caption、类别、审美与安全反馈，以及来源、分辨率、裁剪、许可和覆盖范围等元数据。Pixel 模型读取原始像素；latent 系统还要版本化 encoder、decoder 或 codebook。

训练管线按目标构造实例：VAE 产生重建与 KL 目标，diffusion 采样时间和噪声，AR 构造 pixel 与 code 前缀，GAN 组织真实与生成样本，normalizing flow 计算样本似然。条件生成还要记录 caption 来源、文本编码器与条件丢弃策略。

6.2.5 Benchmark

图像系统需要分别评估分布质量、样本覆盖、文本对齐和人类感知质量。FID 与 precision/recall 衡量生成分布的不同方面；likelihood 适用于能够计算密度的模型；CLIP 类指标和人类评价用于文本条件与视觉质量。

文字渲染、空间关系、计数、身份保持、编辑一致性和安全需要单独测试。比较 pixel 与 latent 方法时，应固定图像分辨率、采样步数、guidance、条件编码器和计算预算，并报告延迟、吞吐、显存及解码器重建误差。

7 结语：五项分析问题

面对一个具体系统，可以把四层框架展开为五项分析问题，将数据表示从架构中单独列出：

1. **数据如何表示？**写出 token、pixel、latent、code 及其转换方式。
2. **它建模什么？**写出概率对象、局部预测目标、状态和生成过程。
3. **它如何计算？**找出信息交互、可见性、状态、复杂度和推理过程。
4. **它从哪里学习？**检查 Database、Dataset、配比、时间、标签和污染。
5. **证据支持什么主张？**检查 Benchmark 的任务、协议、指标、预算和失败案例。

这五问让许多争论变得清楚。“Transformer 和 diffusion 谁更好”把架构与建模混在了一起；“模型在 Benchmark 上接近满分，所以能力已解决”忽略了测量边界；“加更多高质量数据就行”没有说明何为质量、目标分布是什么。

AI 的进展往往来自多个层次共同移动：新的表示改变可建模的数据对象，新的数学接口打开能力空间，新的架构让它可扩展，Data 为训练提供决定参数更新的经验，Benchmark 为研究者提供决定下一步投入的反馈。Data 位于模型训练的中心，Benchmark 通过研究选择塑造领域的演进方向。

> **本节结论**
>
> 五项分析问题构成一套具体的系统阅读方法：表示定义模型看到的数据，建模定义概率对象，架构定义计算结构，Database 定义训练经验与参数更新，Benchmark 定义能力证据并影响研究优化方向。

A 术语表

术语	简明定义
Autoregressive (AR)	用固定顺序把联合分布分解为一系列条件分布。
Benchmark	由任务、测试实例、推理协议、指标和报告规则共同构成的测量契约。
Calibration	模型给出的概率与实际正确频率是否一致。
Dataset	经过采样、标注、清洗与组织的经验集合；同时定义分布与任务边界。
Diffusion model	通过前向破坏和反向去噪路径学习数据分布的生成模型。
Energy-based model (EBM)	用未归一化能量函数为样本赋分，并通过低能量表示高偏好或高概率区域。
Flow matching	通过回归概率路径的速度场来学习连续分布输运。
Foundation model	在广泛数据上训练、可适配多种下游任务的大规模模型。
Generative adversarial network (GAN)	通过生成器与判别器的对抗目标学习隐式样本生成过程。
Inductive bias	模型在有限数据下偏好某类函数或结构的先验倾向。
LLM-as-a-judge	使用语言模型按 rubric 评价开放式模型输出的方法。
Masked diffusion	在离散数据上逐步掩码，并通过反向过程迭代恢复 token 的生成模型。
Mixture of Experts (MoE)	由路由器为输入选择少量专家的稀疏计算层。
Normalizing flow	通过可逆变换和变量替换公式构造显式概率密度的生成模型。
Policy	给定状态或观测时的动作分布。
Policy network	策略的神经网络参数化，在给定状态时输出动作分布或动作。
Quantization	用低比特数值近似权重、激活或缓存，以降低存储、带宽和计算成本。
Score function	概率密度对输入的对数梯度 $\nabla_x \log p(x)$。
State Space Model (SSM)	通过潜在状态随序列演化来表示长程依赖的模型。
Value network	价值函数的神经网络参数化，估计状态或状态-动作对的期望累计回报。
Variational autoencoder (VAE)	通过潜变量生成模型、近似后验与变分下界进行训练。
World model	学习环境状态、观测或奖励如何随动作变化的内部模型。

B 关键公式索引

1. 参数化函数：$\hat{y} = f_\theta(x)$，见equation (1)。
2. 经验风险最小化：$\hat{\theta} = \arg\min_\theta \frac{1}{n}\sum_i \ell_i + \lambda\Omega$，见equation (3)。
3. 类别分布的最大似然：$\hat{\theta} = \arg\max_\theta \sum_i \log[f_\theta(x_i)]_{y_i}$，见equation (4)。
4. AR 链式分解：$p(x) = \prod_t p(x_t \mid x_{<t})$，见equation (9)。
5. 掩码扩散：$q(x_t^i \mid x_0^i) = \alpha_t \delta_{x_0^i} + (1-\alpha_t)\delta_{[\mathtt{MASK}]}$，见equation (15)。
6. 扩散加噪：$x_t = \alpha_t x_0 + \sigma_t \epsilon$，见equation (12)。
7. flow ODE：$dx_t/dt = v_\theta(x_t, t)$，见equation (21)。
8. 强化学习目标：$J(\theta) = \mathbb{E}[\sum_t \gamma^t r_t]$，见equation (6)。
9. 动作价值与优势：$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$，见equation (7)。
10. 缩放点积注意力，见equation (23)。

公式索引用于检查概念是否被完整定义。例如，一篇论文使用“diffusion”这一名称时，还应说明破坏过程、局部预测对象与采样路径。

C 学习资源与常用网站

以下入口按用途精选，均可直接点击；链接于 2026 年 7 月 27 日核对。

类别	常用入口	用途
基础	Python Tutorial；NumPy Quickstart；Mathematics for ML	补 Python、数组计算、线性代数、微积分和概率。
教材	《动手学深度学习》；Deep Learning；Probabilistic ML	分别侧重代码实践、系统理论和概率建模。
课程	MIT 6.S191；CS231n；CS224N；CS336	从总览进入视觉、语言与语言模型训练系统。
实现	PyTorch Tutorials；HF LLM Course；HF Diffusion Course；Colab	完成 tensor、autograd、Dataset、训练与生成实验。
可视化	TensorFlow Playground；CNN Explainer；Netron；Distill	观察决策边界、卷积结构、模型计算图和交互式解释。
论文	arXiv；OpenReview；Semantic Scholar；HF Papers	查论文版本、公开评审、引用关系和近期工作。
Data 与评测	HF Datasets；HELM；MLCommons；AI Index	查 Dataset 工具、模型评测、系统 Benchmark 与年度趋势。

建议顺序

1. 用《动手学深度学习》和 PyTorch 完成一个小型分类器。
2. 按方向选择 CS231n（视觉）或 CS224N（语言）。
3. 学习生成模型时选择 HF Diffusion Course 或 CS336，并明确写出概率对象、训练目标和采样过程。
4. 阅读论文时同时检查代码、Data、Benchmark、版本和评审记录。

参考文献

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2107.03006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.nips.cc/paper/5423-generative-adversarial-nets.

Charles A. E. Goodhart. Problems of monetary management: The u.k. experience. *Monetary Theory and Practice*, pages 91–121, 1984.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. URL https://arxiv.org/abs/2312.00752.

David Ha and Jurgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1807.03039.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu-Jie Huang. A tutorial on energy-based learning. *Predicting Structured Data*, pages 191–246, 2006. URL http://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015. doi: 10.1038/nature14539.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2205.14217.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022. URL https://arxiv.org/abs/2211.09110.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. URL https://arxiv.org/abs/2502.09992.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2103.14749.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.09748.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025. URL https://arxiv.org/abs/2501.14249.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, volume 37, pages 1530–1538, 2015. URL https://arxiv.org/abs/1505.05770.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.07524.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt,

Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020. doi: 10.1038/s41586-020-03051-4.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. URL https://arxiv.org/abs/2210.08402.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1701.06538.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Stanford Institute for Human-Centered Artificial Intelligence. The 2026 ai index report: Technical performance. Stanford University, 2026. URL https://hai.stanford.edu/ai-index/2026-ai-index-report/technical-performance. Accessed 2026-07-13.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016. URL https://proceedings.mlr.press/v48/oord16.html.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1711.00937.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *Proceedings of the Association for Computational Linguistics*, pages 13484–13508, 2023. URL https://arxiv.org/abs/2212.10560.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.07972.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.16502.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://arxiv.org/abs/2306.05685.