第二部分

推荐系统模块

第 4 章 召回与过滤模块

在前面几章中，我们从宏观视角对推荐系统进行了系统性介绍。首先回顾了推荐系统的发展历程、核心目标以及所面临的主要挑战，帮助读者建立对推荐系统整体框架的初步认知；随后进一步介绍了工业界推荐系统的典型架构，包括多业务混合推荐系统架构、单业务级联式推荐系统架构、降级推荐系统以及作者侧推荐系统等内容；最后，我们还讨论了推荐系统中的数据体系、日志体系以及评价体系，帮助读者理解推荐系统背后的数据流与业务闭环。

然而，一个工业级推荐系统往往由多个复杂模块协同组成，仅从整体架构层面理解推荐系统仍然是不够的。从本章开始，我们将采用“庖丁解牛”的方式，按照推荐链路中的执行顺序，对级联式推荐系统的各个核心模块进行逐层拆解，从宏观架构逐步深入到具体算法与工程实现细节，帮助读者理解每一个模块在推荐链路中的职责、原理与设计思想。

本章首先介绍推荐系统链路最上游的两个核心模块：**召回模块（Retrieve）**与**过滤模块（Filtering）**。其中，召回模块主要负责从海量物品库中获取候选内容，而过滤模块则负责对候选结果进行规则约束与质量控制。二者共同构成了推荐链路的起点，并为后续粗排、精排以及重排等模块提供输入数据。

理解召回与过滤模块，不仅有助于读者掌握工业级推荐系统的整体运行逻辑，也能够帮助大家理解后续粗精排、重排以及混排等模块为何要以当前的方式进行设计。因此，在正式介绍各种召回算法与过滤策略之前，我们首先来回答一个最基础但又十分重要的问题：

笔记 推荐系统为什么需要召回？

4.1 为什么需要召回

在初次接触推荐系统时，很多读者都会产生一个直观的想法：

笔记 既然排序模型能够预测用户对物品的兴趣程度，那么为什么不直接利用排序模型对所有物品进行打分，然后选择得分最高的 Top K 个物品推荐给用户呢？

从理论上来看，这种方式确实能够得到全局最优的排序结果；但在工业级推荐系统中，这种做法几乎无法落地。原因在于，推荐系统不仅要追求推荐效果，还必须满足**严格的实时性要求**。前面章节已经介绍过，工业级推荐系统通常直接承载用户请求，整个推荐链路的响应时间往往只有数百毫秒。当用户不断刷新首页、滑动短视频或者浏览商品列表时，系统需要在极短时间内完成一次完整的推荐流程。如果推荐服务耗时过长，比如推荐系统整体耗时超过 1 秒，不仅会增加用户等待时间，在高峰时段甚至可能导致页面卡顿、内容加载缓慢等问题，从而直接影响用户体验。

与此同时，现代互联网平台所拥有的内容规模往往极其庞大。短视频平台可能拥有数亿条视频内容，电商平台可能拥有数千万乃至上亿个商品，新闻资讯平台也可能积累了海量历史文章。如果每次用户请求都对全部候选物品执行复杂排序模型推理，即使单个物品的计算耗时极低，整体计算开销仍然会达到难以接受的程度。换句话说，推荐系统实际上面临着一个典型的矛盾：

- 一方面，需要从海量候选物品中找到用户真正感兴趣的内容；
- 另一方面，又必须在极短时间内完成整个推荐过程。

这本质上是一个在效果与效率之间进行平衡的问题。如果只追求效果，可以对全量物品进行复杂排序；如果只追求效率，则可以随机返回少量内容。然而工业级推荐系统必须同时兼顾二者，因此问题就变成了：

笔记 如何在数百毫秒甚至更短的时间内，从亿级候选物品中筛选出用户最可能感兴趣的少量内容？

从某种意义上来说，这无异于在极短时间内完成一次“大海捞针”。而召回模块的出现，正是为了解决这一问题。为了实现这一目标，工业界逐渐演化出了今天广泛采用的**级联式（Cascade）推荐架构**。在该架构中，召回模块位于整个推荐链路的最上游，其核心职责是利用高效率的检索策略快速缩小候选物品池的范围，将候选规模从千万级、亿级压缩至万级甚至千级，从而为后续排序阶段创造计算空间。

在这种架构下，推荐链路被拆分为多个阶段：召回、粗排、精排以及重排。每个阶段负责解决不同规模下的筛选问题。其中，召回模块位于整个推荐系统的最上游，其核心任务如下：

定义 4.1

召回模块需要利用相对简单且高效的策略，从海量候选池中快速筛选出一批用户可能感兴趣的候选物品，将候选规模从千万级、亿级缩减至万级甚至千级，从而为后续排序阶段创造计算空间。

需要注意的是，召回模块承担的是**“大范围筛选”的职责**，因此其设计目标并不是获得最精准的排序结果，而是在尽可能短的时间内保证候选集的覆盖率与召回率。这也决定了召回阶段所采用的模型和策略通常不能过于复杂，否则其本身就会成为推荐链路的性能瓶颈。正因如此，工业界发展出了协同过滤召回、Embedding 召回、双塔召回、图网络召回等一系列高效率召回技术，并以并行执行的方式作用于多路召回架构中。而这些内容也将是本章后续重点讨论的对象。

4.2 多路召回架构

在上一节中我们介绍了召回模块存在的必要性：推荐系统需要在极短的时间内，从百万级、千万级甚至亿级物品中快速圈定用户可能感兴趣的候选集合。因此，工业界几乎所有推荐系统都会在推荐链路的最上游部署召回模块，通过较低的计算成本完成候选集的第一次大规模筛选。

需要注意的是，召回模块的核心目标并不是获得最精准的推荐结果，而是在保证极低延迟的前提下，尽可能提高候选集的覆盖率与召回率。因此，工业界很少依赖单一召回策略，而是普遍采用**多路召回（Multi-channel Retrieve）架构**：通过并行执行多种不同的召回策略，从不同角度挖掘用户潜在兴趣，再将各路结果进行汇总，从而兼顾个性化、多样性以及业务目标。

从工程实现角度来看，多路召回架构通常可以划分为三个阶段：**召回前处理（Pre-retrieval Processing）**、**多路召回执行（Multi-channel Retrieval Execution）**以及**召回后处理（Post-retrieval Processing）**。整体流程如图4.1所示，下面将会为大家详细介绍召回模块内部的具体处理流程。

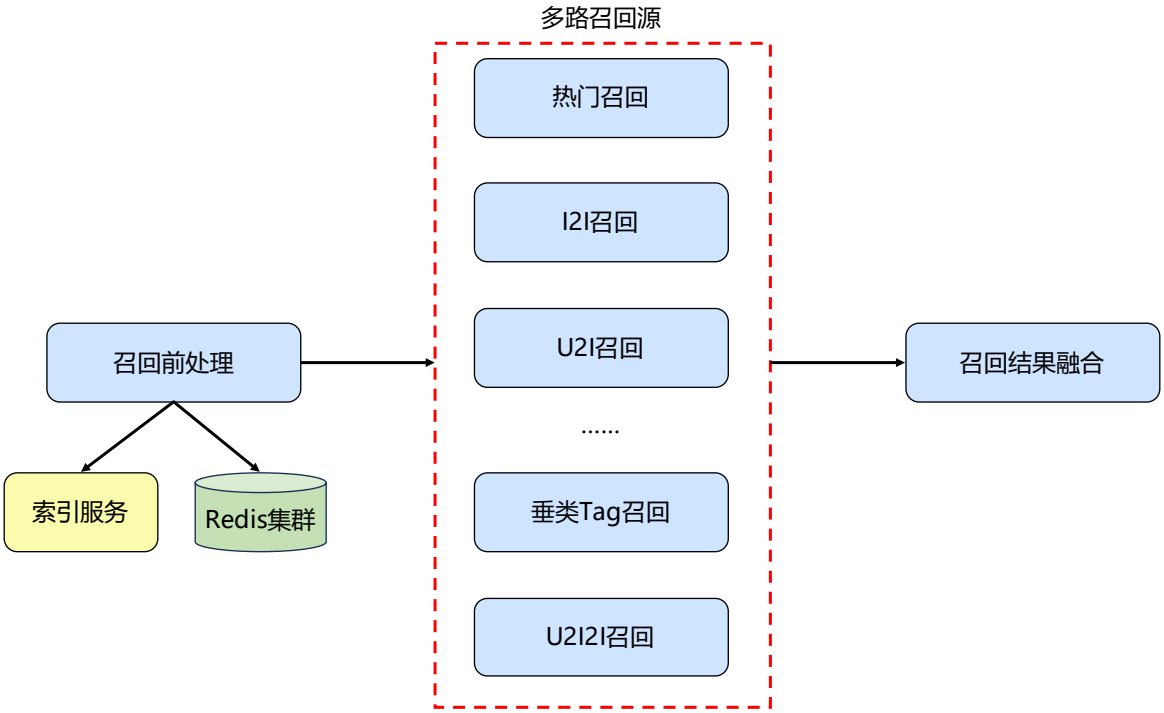

图 4.1: 召回模块内部处理流程。

召回前处理

在正式发起召回请求之前，推荐系统首先需要完成一系列前置准备工作，为后续各路召回提供必要的数据支持。该阶段通常包括以下几个步骤：

- **构建召回触发信号（Trigger）**

  召回系统并不会直接使用全部用户行为，而是需要从用户近期行为中提取能够代表当前兴趣的触发信号。例如用户最近点击、浏览、点赞、收藏或购买过的物品，以及用户近期关注的话题标签、关键词等信息。这些触发信号将作为后续召回的输入。

- **获取辅助特征**

  为了提升召回效果，系统通常还会读取用户画像、上下文环境以及设备信息等辅助特征。例如年龄、性别、兴趣标签、地理位置、访问时间、设备机型等信息。

- **访问外部存储服务**

  召回阶段通常需要访问 Redis、索引服务以及用户画像服务等外部系统。例如读取用户最近交互的物品列表、查询用户长期兴趣标签，或者根据特定触发项获取对应的候选物品集合。

在工业级推荐系统中，召回所依赖的索引体系通常分为两类：

- **正排索引（Forward Index）**

  以 Item ID 为 Key 存储物品完整特征，例如标题、标签、统计特征以及 Embedding 向量等信息。其主要解决的问题是：

笔记 已知 Item ID，如何快速获取其完整特征？

- **倒排索引（Inverted Index）**

  以用户、物品、标签、关键词、类目等触发项作为 Key，映射到一组关联物品列表。其主要解决的问题是：

笔记 已知某个兴趣触发项，如何快速找到相关物品？

  在工程实践中，每个触发项对应的物品集合通常被称为**倒排拉链（Inverted List）**。例如某个视频对应的相似视频集合、某个兴趣标签对应的内容集合，均属于典型的倒排拉链结构。

多路召回执行

完成前处理后，系统便进入多路召回执行阶段。该阶段是整个召回模块的核心部分，其主要特点包括：多种召回策略并行执行，每一路召回从不同角度刻画用户兴趣，各路召回结果最终汇总形成统一候选池。工业界常见的召回方式包括：

- **热门召回（Hot Retrieval）**

  根据全站或垂类热度统计结果进行召回，用于保障基础流量覆盖以及新用户冷启动。

- **I2I 召回（Item-to-Item Retrieval）**

  基于物品协同过滤（Item CF）或物品 Embedding 相似度，根据用户历史交互物品寻找相似内容。

- **U2I 召回（User-to-Item Retrieval）**

  通过用户向量与物品向量进行近邻搜索实现召回，通常由双塔模型生成 Embedding，再结合 ANN 检索系统完成召回。

- **垂类召回（Tag Retrieval）**

  基于用户兴趣标签、内容标签或类目进行匹配召回。

- **U2U2I 召回（User-to-User-to-Item Retrieval）**

  先寻找相似用户，再聚合相似用户喜欢的物品进行推荐，本质上属于 User CF 思想。

- **U2I2I 召回（User-to-Item-to-Item Retrieval）**

  在用户历史行为基础上进行两跳扩展，通过相似物品链路增强候选集多样性。

- **实时行为召回（Real-time Retrieval）**

  基于用户当前会话中的即时行为动态触发召回，例如刚刚点击的视频、搜索关键词等。

- **图网络召回（Graph Retrieval）**

利用用户-物品交互图学习高阶关系，通过图神经网络生成 Embedding 后进行向量检索。

由于各路召回需要同时面对大规模候选库，因此现代推荐系统通常采用异步 RPC、多线程调度或微服务并发调用等方式实现并行执行，从而将整体召回耗时控制在几十毫秒以内。

召回后处理

当所有召回源返回结果后，系统会进入召回后处理阶段，对候选集合进行统一整理。该阶段主要包括以下几个操作：

- **去重（Deduplication）**
  由于同一物品可能同时出现在多个召回源中，因此需要根据 Item ID 进行统一去重。
- **轻量级过滤（Lightweight Filtering）**
  过滤掉已经下架、违规、失效或用户明确屏蔽的物品。
- **Quota 控制与截断（Quota & Truncation）**
  为了保证后续排序模块的计算开销可控，系统通常会对候选规模进行限制。例如为每个召回源设置独立配额 (Quota)，再对整体候选集进行截断，将候选规模控制在数百至数千个物品范围内。

需要注意的是，此处的过滤主要是为了保证召回结果的有效性和可用性，因此通常只包含一些简单且计算成本较低的规则。而涉及复杂业务逻辑、多样性控制、公平性约束、生态调控以及风险治理等操作，则通常会在后续独立的**过滤模块（Filtering）**中统一处理。这样的设计能够保证召回模块始终保持高吞吐、低延迟的特点，同时也使整个推荐链路具备更好的模块化与可扩展性。

4.3 召回技术分类

在上一节中，我们介绍了工业级推荐系统中的多路召回架构。需要注意的是，多路召回本质上是一种**工程组织方式**，它解决的是“如何将多种召回策略高效协同起来”的问题；而不同召回策略背后所采用的算法与技术路线，则属于**召回技术范畴**。

事实上，一个成熟的工业级推荐系统通常同时运行着数十甚至上百路召回策略。例如，热门召回、协同过滤召回、双塔召回、图召回、标签召回等都可能在同一次推荐请求中并行执行。虽然这些召回路在触发方式、服务形态和业务目标上存在差异，但从底层原理来看，大多数都可以归纳为少数几种经典的技术范式。因此，在深入学习具体召回模型之前，有必要先从整体上建立工业界召回技术的知识框架。

从推荐系统的发展历程来看，召回技术经历了明显的演进过程：

$$\text{规则驱动} \rightarrow \text{协同过滤驱动} \rightarrow \text{表征学习驱动} \rightarrow \text{图关系建模驱动}$$

在推荐系统发展的早期阶段，由于用户行为数据较少，系统主要依赖热门推荐、标签匹配、运营规则等人工策略进行内容分发；随后，随着用户行为数据的大规模积累，协同过滤（Collaborative Filtering）逐渐成为推荐系统的核心技术，通过挖掘用户与物品之间的共现关系实现个性化推荐；近年来，深度学习的发展进一步推动了召回技术的升级，Embedding、双塔模型、多兴趣建模等方法使系统能够学习更丰富的用户兴趣表示；而图神经网络（Graph Neural Network, GNN）的引入，则使推荐系统能够利用用户、物品以及多种实体之间的高阶关联关系，从更复杂的图结构中挖掘潜在兴趣。

总体而言，当前工业界主流的召回技术大致可以分为以下四类：

1. 协同过滤召回（Collaborative Filtering Retrieval）
2. 内容与规则召回（Content & Rule-based Retrieval）
3. 向量召回（Embedding-based Retrieval）
4. 图召回（Graph-based Retrieval）

需要说明的是，这种分类方式是按照**技术实现原理**进行划分的。而在工程实践中，我们还经常会看到诸如 U2I、I2I、A2A，U2U2I、U2I2I 等命名方式。这二种划分方式其实并不冲突。前者关注的是**召回策略采用什么技术实现**；后者关注的是**召回结果是沿着什么关系路径产生的**。例如：

- Item-CF 属于协同过滤召回，同时也是典型的 I2I（Item-to-Item）召回；
- User-CF 属于协同过滤召回，同时也是典型的 U2U2I（User-to-User-to-Item）召回；
- 双塔模型召回属于向量召回，同时也是典型的 U2I（User-to-Item）召回；
- 图神经网络召回既可以实现 U2I，也可以实现 I2I、U2U2I、U2I2I 等多种召回路径。

从关系路径的角度来看，工业界较为常见的召回链路包括：

- **U2I（User-to-Item）**：基于用户兴趣直接寻找潜在感兴趣的物品；
- **I2I（Item-to-Item）**：基于用户已交互物品寻找相似物品；
- **A2A（Author-to-Author）**：基于创作者关系寻找相似创作者；
- **U2U2I（User-to-User-to-Item）**：先寻找相似用户，再推荐相似用户喜欢的物品；
- **U2I2I（User-to-Item-to-Item）**：先找到用户历史交互物品，再扩展到相似物品；
- **I2U2U（Item-to-User-to-User）**：先找到与物品交互过的用户，再扩展到相似用户；
- **I2I2U（Item-to-Item-to-User）**：先找到相似物品，再寻找与这些物品交互的用户。

其中，U2U2I 与 U2I2I 是用户侧推荐系统中最常见的两类召回模式；而 I2U2U 与 I2I2U 则更多应用于作者侧推荐系统（Creator Recommendation）中，用于寻找潜在感兴趣用户，从而实现内容分发与创作者成长扶持。

上述几种典型召回路径的关系如图 4.2 所示。从本质上看，这些路径中的“相似用户”和“相似物品”模块往往可以复用同一套相似度计算服务，例如点积相似度检索、协同过滤、Embedding 检索等，因此它们更多体现的是推荐信号传播的路径差异，而非底层技术实现的差异。

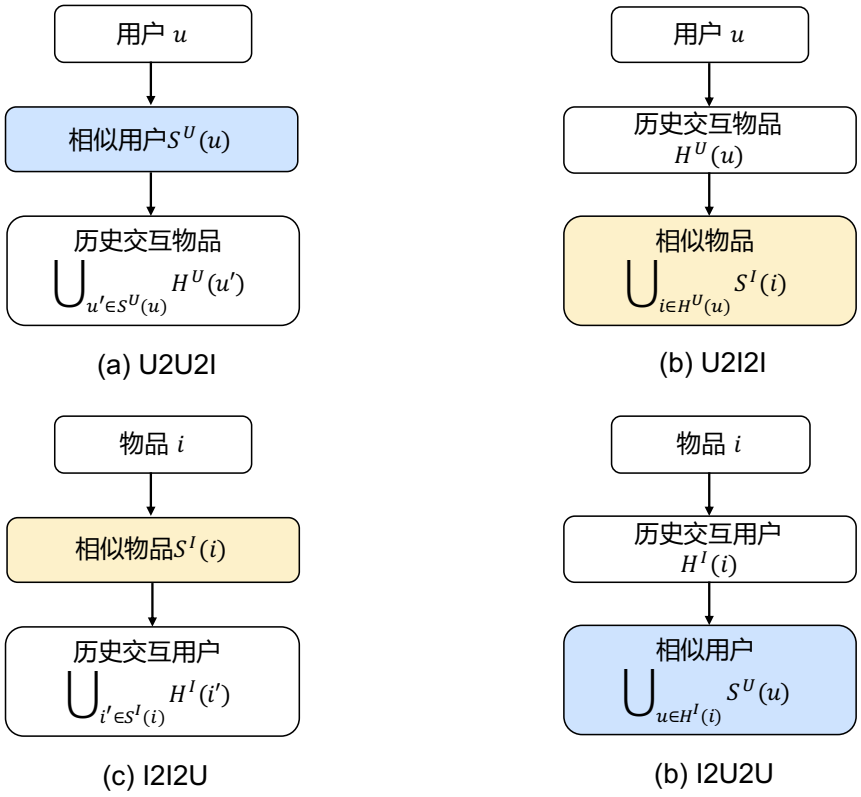

图 4.2: U2U2I、U2I2I、I2I2U、I2U2U 召回方案对比。

本书后续将主要按照**技术路线**展开介绍，而将 U2I、I2I 等关系路径作为理解具体策略的重要辅助视角。这样既能够帮助读者理解推荐系统技术的发展脉络，也有助于在实际工程中快速定位不同召回策略的作用与实现方式。四类主流召回技术的整体对比如表 4.1 所示。接下来，我们将分别介绍协同过滤召回、内容与规则召回、向量召回以及图召回四种典型技术范式，并进一步分析其核心思想、工程实现方式及适用场景。

表 4.1: 主流召回技术分类对比

召回类型	核心技术	典型路径	优点	缺点	适用场景
协同过滤召回	User-CF、Item-CF、Swing	I2I、U2U2I	简单高效、可解释性强	冷启动能力较弱	行为数据丰富的平台
内容与规则召回	Tag 匹配、热门推荐、运营规则	Tag、Hot、Category	实现简单、可控性强	个性化能力有限	冷启动与运营场景
向量召回	双塔、MIND、多兴趣模型	U2I	泛化能力强、效果优秀	依赖模型训练	主流个性化推荐系统
图召回	GNN、随机游走、图 Embedding	U2I2I、U2U2I	能够建模高阶关系	工程复杂度较高	复杂关系网络场景

4.3.1 协同过滤召回

协同过滤（Collaborative Filtering，CF）是推荐系统历史上最经典的一类召回技术，也是工业界最早实现大规模个性化推荐的重要方法。协同过滤的核心思想可以概括为：

定义 4.2

喜欢相似内容的用户，其未来兴趣往往也相似；与某个物品经常一起被消费的物品，未来也可能被共同消费。

与基于内容理解的方法不同，协同过滤并不关心物品本身的语义信息，而是完全依赖用户行为数据所形成的共现关系进行推荐。根据建模对象的不同，协同过滤通常又可进一步分为：User-CF（User Collaborative Filtering）、Item-CF（Item Collaborative Filtering）、Swing 召回等。其中，User-CF 属于典型的 U2U2I 路径，Item-CF 属于典型的 I2I 路径，Swing 则是在 Item-CF 基础上的工业级改进方案。

由于协同过滤具有实现简单、可解释性强、线上性能开销低等优势，即使在深度学习广泛应用的今天，依然是工业界召回体系中的重要组成部分。在后续第10章中，我们将重点介绍 User-CF、Item-CF 与 Swing 的具体算法原理。

4.3.2 内容与规则召回

内容与规则召回（Content & Rule-based Retrieval）是工业界应用最广泛的基础召回策略之一。与协同过滤依赖用户行为不同，这类召回方法更多利用物品属性、用户画像以及业务规则来完成候选集构建。典型的内容与规则召回策略包括：热门召回（Hot Recall）、Tag 召回（Tag Recall）、类目召回（Category Recall）、地域召回（Location Recall）、运营召回（Operation Recall）、新内容召回（Fresh Recall）等。例如：

- 科技兴趣用户召回科技类内容；
- 北京地区用户优先召回北京本地服务；
- 新用户优先召回全站热门内容；
- 运营活动期间优先召回活动内容。

内容与规则召回最大的优势在于其可控性与稳定性较强，因此经常承担冷启动扶持、运营干预、内容保量等职责。尽管这类方法个性化能力相对有限，但在工业级推荐系统中仍然是不可或缺的重要组成部分。内容/规则召回通常不依赖模型，而是一种纯业务导向的推荐策略，在工业界应用中较为常见。一般来说，召回方法在实际落地时都包含离线部分和在线部分：离线部分主要涉及数据的统计分析或模型的训练；在线部分则通常指模型的实时推理或召回策略的即时执行。以 Tag 召回为例，这种策略导向的召回方式具体实现如下图4.3所示。在离线阶段，系统以用户偏好的兴趣标签（Tag）作为触发信号，通过离线计算任务预先产生每个兴趣 Tag 对应的候选物品列表。这些物品的选择可以基于物品的后验表现，例如曝光量 Top K、点赞数 Top K、评论数 Top K 等指标；同时，也可以引入一定比例的随机采样物品，以避免候选集过度集中于热门内容，从而缓解因推荐系统自身偏差（bias）导致的同质化问题。完成离线计算后，系统将每个兴趣 Tag 及其对应的物品列表以键值对（K-V）

的形式存储在 Redis 缓存中。在线上阶段，系统根据用户当前的多个兴趣偏好 Tag，直接从 Redis 中查询对应的物品列表，并将多个 Tag 的召回结果进行合并，即可高效完成整个在线召回过程。

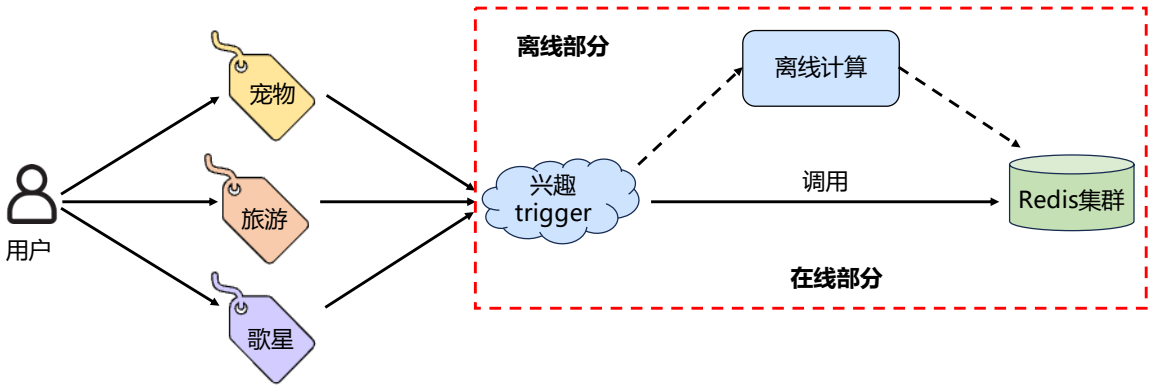

图 4.3: 用户兴趣 Tag 召回示意图。

4.3.3 向量召回

随着深度学习技术的发展，向量召回（Embedding-based Retrieval）逐渐成为当前工业界最主流的召回范式。其核心思想是：

定义 4.3

将用户与物品映射到同一个向量空间中，通过向量相似度衡量用户与物品之间的匹配程度。

在离线阶段，系统利用海量用户行为数据训练 Embedding 模型，学习用户向量与物品向量；在线阶段，则通过近似最近邻搜索（Approximate Nearest Neighbor, ANN）快速检索与用户向量最接近的物品集合。目前工业界常见的向量召回模型包括：DSSM、YouTube DNN、双塔模型（Two-Tower）、MIND、ComiRec、SDM、PDN 等。

相比传统协同过滤方法，向量召回能够融合用户画像、内容特征、上下文信息等多维特征，具备更强的泛化能力与冷启动能力，因此已经成为当前推荐系统召回层的核心技术。

4.3.4 图召回

从本质上来看，**推荐系统天然可以表示为一个大规模用户-物品二分图**。用户与物品之间的点击、浏览、购买、点赞等行为构成图中的边，而用户和物品则构成图中的节点。

图召回（Graph-based Retrieval）正是利用这种图结构信息进行推荐。与传统协同过滤主要利用用户与物品之间的一阶共现关系不同，图召回能够进一步挖掘用户与物品之间的二阶、三阶乃至更高阶关联关系。例如，用户 A 喜欢物品 X；物品 X 又被用户 B 喜欢；而用户 B 进一步喜欢物品 Y。那么即使用户 A 从未与物品 Y 产生过直接交互，系统仍然可以通过这条关联链路推断出用户 A 可能对物品 Y 感兴趣，并将其纳入候选集合。

这种基于图结构的高阶兴趣传播能力，使得图召回能够发现大量隐含的用户兴趣关联关系，从而突破传统协同过滤仅依赖局部共现信息的限制。在图学习领域，这类由多个节点和边依次连接形成的关联轨迹通常被称为**路径（Path）**；而在异构图（Heterogeneous Graph）场景下，当路径中包含不同类型的节点与关系时，则进一步称为**元路径（Meta-path）**。通过对这些高阶路径进行建模，图召回能够更充分地利用用户、物品、作者、类别、标签等多种实体之间的复杂关系，从而提升推荐结果的相关性与泛化能力。当前工业界常见图召回技术包括：DeepWalk、Node2Vec、GraphSAGE、PinSage、LightGCN、HGT 等。图召回通常被视为协同过滤与深度学习结合后的进一步演进方向，在社交推荐、内容推荐、电商推荐等场景中均有广泛应用。

本节从技术实现角度对工业界主流召回方法进行了分类介绍。总体来看，协同过滤召回侧重挖掘显式行为

共现关系，内容与规则召回强调业务可控性与冷启动能力，向量召回通过表示学习实现深层兴趣建模，而图召回则进一步利用用户与物品之间的高阶关联关系提升召回效果。

这些技术并非相互替代，而是在工业级推荐系统中以多路召回的形式协同工作，共同构成完整的召回体系。后续章节中，我们将首先从推荐系统最经典的协同过滤召回开始，逐步深入分析各类召回模型的原理与工程实践。

4.4 召回结果融合

在工业级推荐系统中，多路召回模块通常会并行执行数十甚至上百种不同的召回策略。每一路召回源都会独立返回一批候选物品，例如，热门召回、协同过滤召回、双塔召回、图召回以及各种业务策略召回等。然而，多路召回执行完成后得到的结果并不能直接进入后续排序模块。这是因为不同召回源之间往往存在大量重叠物品，同时各路召回返回的候选规模也存在较大差异。如果缺乏统一的融合机制，容易导致候选集规模失控、候选分布失衡以及系统稳定性下降等问题。

因此，在召回阶段结束之后，通常会引入一个专门的**召回结果融合模块（Retrieve Fusion Module）**，负责对所有召回源的结果进行统一处理。其核心目标包括：

- 保证候选集规模满足后续排序模块要求；
- 保证不同召回源之间的合理流量分配；
- 提升候选集的覆盖率与多样性；
- 提高系统整体稳定性与鲁棒性。

典型的召回融合流程如图4.4所示，通常包括去重、配额控制、截断、兜底等多个步骤。

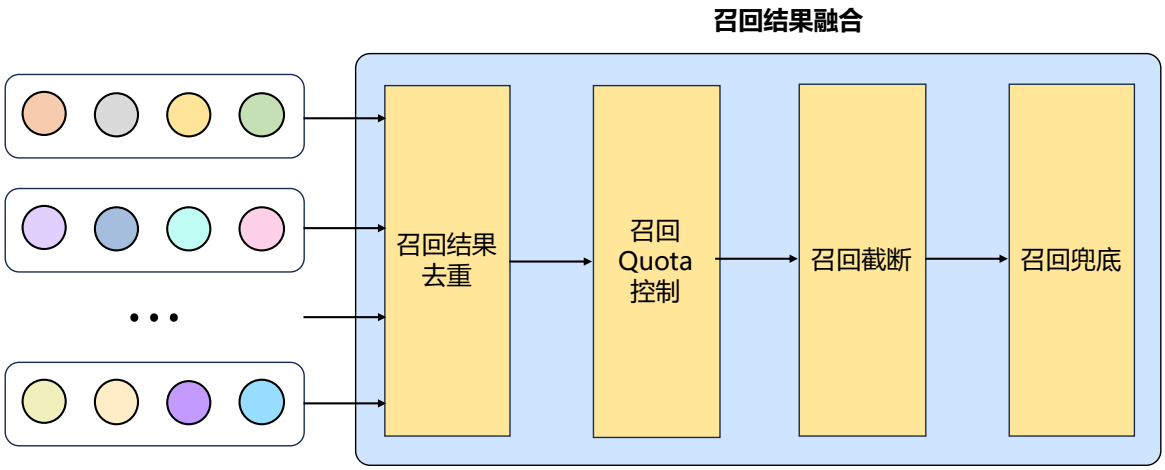

图 4.4: 召回结果融合内部处理流程。

召回结果去重

由于不同召回源往往基于不同的策略寻找候选物品，因此同一个物品可能同时出现在多个召回源中。例如：某个热门视频可能同时被热门召回与双塔召回召回，某个商品可能同时出现在 Item-CF 召回与图召回结果中，某个作者作品可能同时命中标签召回与向量召回。如果不进行去重处理，则会导致候选集中出现大量重复物品，不仅浪费后续排序模型的计算资源，还会降低召回覆盖率。因此，工业界通常会基于物品 ID 对所有召回结果进行**统一去重（Deduplication）**。

需要注意的是，在多路召回中，一个物品可能被多个召回源同时命中。例如，$item_i \rightarrow \{CF, DSSM, GNN\}$ 表示该物品同时被协同过滤、双塔模型以及图召回三种召回策略召回。在召回结果去重之后，系统通常会为每个物品保留一个**召回来源标识（retrieval reason）**。该标识在后续粗排、精排以及线上监控、问题回溯等环节都

具有重要作用，也可以作为排序模型的重要输入特征。对于同一个物品被多路召回同时命中的情况，召回 reason 字段通常只记录其中一路召回来源，而不会同时保存所有召回源。例如，Item ID 为 $i257s4v9$ 的物品，其召回 reason 可以记录为 5536，对应某一路具体的召回策略。关于召回 reason 设置为何种召回源，工业界通常有两种常见做法：一种是按照召回源优先级进行选择，例如热门召回优先于协同过滤召回，协同过滤召回优先于双塔召回；另一种是按照召回源内部得分进行选择，例如选择得分最高的召回源作为最终的 reason，但这种方式可能会存在不同召回源打分不太可比的问题。无论采用哪种方式，最终都需要保证每个物品在候选集中的唯一性。由于召回 reason 通常只是作为一种标识，本身不会带来线上 AB 指标的收益，所以工业界通常会选择一种简单且易于实现的方式进行设置。

召回源配额控制

不同召回策略返回的候选规模往往差异巨大。例如：热门召回可能返回上万个候选，双塔召回可能返回数千个候选，图召回可能仅返回数百个候选，某些垂类召回甚至只有几十个候选。如果简单地将所有召回结果直接合并，则容易导致候选集被少数召回源主导，从而降低整体召回的多样性。

表 4.2: 召回源 Quota 示例

召回源	原始候选数	保留数量
热门召回	10000	500
Item-CF 召回	3000	1000
双塔召回	5000	1500
图召回	1000	1000

因此，工业界通常会为每一路召回源设置独立的 **Quota（召回配额）**，如表4.2所示。例如：热门召回了 1 万个候选物品，Item-CF 召回了 3000 个，双塔和图召回分别召回了 5000 个和 1000 个。但是为了保证粗排阶段输入候选集的大小在 5000 范围以内，所以对于热门召回可以只保留 500 个物品，其他召回分别保留 1000、1500 和 1000 个候选物品。通过这种召回 Quota 控制，可以避免某一路召回源占据过多流量，从而维持候选集的丰富度与多样性。

候选集截断

在完成多路召回融合后，系统通常仍会得到数万级别的候选物品。然而后续粗排和精排模型的计算成本远高于召回阶段，因此不可能对无限规模的候选集进行排序。例如：召回阶段候选数为 50000，粗排输入候选数为 5000，而精排输入候选数为 500。因此，在召回融合阶段通常需要进行**统一截断（Truncation）**。工业界常见的做法包括：按召回源 Quota 截断、按召回源内部得分截断、按业务优先级截断、按动态流量分配策略截断等。最终，系统会将候选规模控制在后续排序模块能够接受的范围内。

召回源兜底机制

在实际线上环境中，任何一个召回服务都有可能因为模型异常、索引失效、RPC 超时或机器故障而返回空结果。例如：双塔向量索引服务异常，Redis 缓存失效，图召回服务调用超时，特征服务不可用等等。如果缺乏保护机制，则可能导致整体召回结果数量急剧下降，进而影响后续排序效果与用户体验。

因此，工业界通常会设计多层次的**召回兜底机制（Backup Mechanism）**。常见的方案包括：热门内容召回兜底，类目热门召回兜底，地域热门召回兜底，Redis 缓存结果兜底，历史推荐结果兜底等。例如，当双塔召回服务超时时，可以直接补充一批热门内容候选，从而保证召回规模不低于预设阈值。虽然这种方式可能降低推荐精度，但能够有效保障推荐系统的基本可用性。

召回监控与质量评估

召回融合模块同时也是推荐系统监控体系的重要组成部分。工业界通常会重点监控以下指标：各召回源召回数量，各召回源覆盖率，各召回源去重率，各召回源点击率（CTR），各召回源观看时长（Duration），各召回源超时率，各召回源空结果率等。通过这些监控指标，算法工程师能够快速发现召回链路中的异常情况，并及时进行流量调整或服务降级。

总体而言，召回结果融合并非简单地将多路召回结果拼接在一起，而是连接召回模块与排序模块的重要桥梁。其本质是在保证系统实时性与稳定性的前提下，尽可能保留不同召回策略所提供的有效信息，为后续排序阶段提供高质量、高覆盖率且规模可控的候选集合。

4.5 过滤限流模块

与召回模块负责从海量候选物品中尽可能找回用户可能感兴趣的内容不同，过滤与限流模块（Filtering & Throttling）承担的是推荐系统中的“守门人（Gatekeeper）”角色。推荐系统本质上是一种流量分配系统，它决定了哪些内容能够被用户看到，哪些内容无法获得曝光。因此，推荐系统不仅需要关注点击率（CTR）、观看时长（Watch Time）等业务指标，还必须保证推荐结果符合平台治理规则、法律法规要求以及内容生态建设目标。

如果缺乏有效的过滤机制，推荐系统可能出现一系列问题。例如，用户已经消费过的内容被反复推荐；违规内容进入推荐链路；低质量内容大量占据流量；黑产账号利用推荐系统获取曝光；甚至导致平台内容生态出现“劣币驱逐良币”的现象。因此，在工业级推荐系统中，过滤与限流模块几乎贯穿整个推荐链路，通过持续筛选和约束候选内容，确保最终推荐结果在用户体验、内容安全、平台生态以及法律合规之间取得平衡。

4.5.1 过滤模块的位置

很多初学者容易认为过滤模块只存在于召回之后，实际上并非如此。在工业级推荐系统中，过滤逻辑通常贯穿整个推荐流程，在不同阶段承担不同职责。在内容进入推荐系统之前，系统通常会进行**召回前过滤（Pre-retrieval Filtering）**。这一阶段主要负责剔除已经下架、审核未通过、版权违规或删除状态的内容，避免无效内容进入后续推荐链路，从而减少系统计算资源浪费。在多路召回完成之后，系统会执行**召回后过滤（Post-retrieval Filtering）**。该阶段主要处理用户已消费内容过滤、用户屏蔽内容过滤、黑名单作者过滤以及地域限制过滤等逻辑，进一步提升候选集质量。进入排序阶段之后，过滤逻辑并不会消失。许多平台会在排序结果基础上增加**排序后过滤（Post-ranking Filtering）**策略，例如广告比例控制、风险内容降权、同作者内容数量限制等，以避免排序模型过度追求点击率而损害用户体验。

为了简化起见，我们这里只介绍召回模块之后统一的过滤模块，对于推荐系统后链路排序环节中的过滤逻辑在此不做过多的赘述。本节主要是讲解过滤限流模块作为一种安全合规的手段，在推荐系统中是如何发挥作用的。

4.5.2 常见过滤与限流策略

从工业实践来看，过滤与限流策略通常来源于多个维度的信息，包括内容安全、用户体验、生态治理以及业务规则等。典型策略如表4.3所示。通常来说，短视频推荐系统中的过滤与限流策略相对数量比较少且逻辑较为简单，而直播推荐系统中的过滤与限流策略数量会很多且逻辑更为复杂。由于直播是一种天然实时传播的内容形态，在安全合规方面的审核难度更大，因此复杂的直播推荐系统通常会有几百甚至上千种复杂的过滤限流策略，进而保证内容平台的安全合规性，防止出现直播内容不当而产生的社会舆情及法律风险。

除了简单的过滤之外，工业界通常还会使用限流（Throttling）机制对部分内容进行曝光控制。例如，一些内容虽然并未达到违规标准，但由于内容质量较低、原创性存疑或用户负反馈较高，系统可能不会直接过滤，而是通过降低其流量配额的方式控制其传播范围。相比完全过滤，限流能够在降低风险的同时减少误伤，从而在内容安全与内容供给之间取得更加平衡的效果。

表 4.3: 过滤限流策略示例

过滤限流维度	典型规则示例	过滤限流策略
封面/画面合规	封面含色情裸露、低俗暗示、血腥暴力、恐怖元素、敏感标识（如国旗误用）等	随机过滤 80% 的请求
标题/文本违规	包含违禁词、夸大误导、政治敏感词、虚假宣传、诱导点击等	随机过滤 80% 的请求
内容主题风险	涉及政治事件、宗教极端、民族歧视、谣言传播、非法集资、敏感社会事件等	100% 完全过滤，内容仅创作者可见
内容质量低下	画面模糊、音画不同步、纯黑屏/白屏、无实质信息、大量水印或广告插入等	随机过滤 75% 的请求，总曝光不能超过 X 次
原创性存疑	内容疑似 AI 批量生成、无真人出镜、脚本高度模板化、存在高概率抄袭或洗稿行为、内容属于二次搬运内容、存在版权违规风险	全域随机过滤 60% 的请求
用户行为反馈	被大量用户举报、低完播率 + 高跳出率组合、负向互动（如踩、拉黑）集中爆发等	不同活跃度人群不同的随机过滤比例
发布者信誉	账号处于黑名单（比如黑产账号）、历史违规频发、新号无认证、批量发布相似内容等	大流量公域页面 100% 完全过滤，小流量页面随机 50% 的请求

4.5.3 负向判罚体系

在实际工业系统中，过滤模块通常并不会直接判断某个内容是否违规，而是更多承担“策略执行器”的角色。违规内容的识别往往由独立的内容安全系统、社区治理系统以及反作弊系统完成，而这些系统产生的风险标签最终统一汇聚到**负向判罚体系（Penalty System）**中。负向判罚体系负责统一接收各种风险信号，例如内容安全标签（涉黄、涉暴、涉政等）、用户举报信息、人工审核结论、账号风险等级、行为异常检测结果、AI 生成内容风险标签等。根据风险等级的不同，系统会采用不同的处罚策略。

对于高风险内容，通常采用**硬过滤（Hard Filtering）**策略，直接阻断其进入任何推荐链路。例如涉黄、涉暴、违法违规内容，其曝光量直接被限制为零。对于中风险内容，系统通常采用**软限流（Soft Throttling）**策略。例如保留部分流量、降低排序分数或者限制分发范围，从而减少其传播影响。对于某些仅对特定用户群体存在风险的内容，则会采用**定向屏蔽（Targeted Blocking）**策略。例如针对未成年人屏蔽成人内容，或针对特定国家和地区限制部分内容展示。此外，还有一种常见策略称为**风险打标（Risk Tagging）**。系统并不直接过滤内容，而是将风险标签透传给排序模型，由排序模型综合考虑内容质量、用户兴趣以及风险等级后共同决定最终曝光概率。这种多层级的负向判罚体系，使得推荐系统能够更加灵活地平衡内容安全与用户体验之间的关系。

4.5.4 风控算法的发展

需要特别说明的是，上述风险标签大多数并非由推荐系统直接生成，而是由专门的风控团队负责建设和维护。推荐系统通常只需要消费这些标签，并依据预设策略执行相应操作，而无需关心标签背后的生成逻辑。

以短视频场景中的违规内容识别为例，风控算法的发展大致经历了两个阶段。在早期阶段，内容审核主要依赖任务专用模型。针对不同风险类型分别训练独立模型，例如色情识别模型、暴力识别模型、OCR 文本识别模型、人脸检测模型以及 Deepfake 检测模型等。常见网络包括 ResNet [3]、MobileNet [4]、EfficientNet [7] 以及 ViT [2] 等。这种方案在单任务场景下往往能够取得较高精度，但存在明显缺点：模型数量众多、维护成本较高，并且当出现新的风险类型时，通常需要重新构建数据集并训练新的模型。

近年来，随着大语言模型（LLM）与多模态大模型（VLM）的快速发展，内容审核逐渐进入统一建模阶段。以 CLIP [6]、BLIP [5]、Qwen-VL [1] 等模型为代表的新一代多模态模型，能够同时理解图像、视频、语音以及文本信息，并将不同模态映射到统一语义空间中。在这一框架下，系统不再需要为每种风险单独训练模型，而是可以通过统一的大模型完成多种违规任务识别。例如，通过图文相似度判断色情内容，通过跨模态语义理解发现隐晦违规内容，通过语言模型生成可解释的审核理由等。这种技术路线显著提升了风控系统的泛化能力和迭代效率，也成为当前内容安全领域的重要发展方向。

值得一提的是，本节所提及的 ViT、CLIP、BLIP、Qwen-VL 等多模态模型，不仅在风控领域发挥关键作用，也正成为推荐算法的新兴前沿技术。我们将在第33章中对这些技术在推荐系统中的应用进行详细探讨。

4.5.5 过滤模块面临的挑战

尽管当前过滤与限流体系已经相对成熟，但仍然面临诸多挑战。首先，AIGC 技术的发展使违规内容生成成本大幅降低，传统规则和小模型越来越难以应对快速演化的新型风险内容。其次，跨模态违规问题日益突出。例如视频画面本身完全合规，但语音内容包含敏感信息；或者标题正常但评论区存在违规引导行为。这些场景都要求系统具备更强的多模态理解能力。此外，过滤策略还面临误伤优质内容的问题。过于严格的过滤机制可能压制优质长尾创作者的发展，而过于宽松的策略又可能损害平台生态。因此，如何在安全性与内容供给之间取得平衡，一直是工业界持续研究的重要课题。

未来，过滤模块将逐步从传统的规则驱动模式演进为融合大模型理解能力的智能治理系统，在保障内容安全的同时，更加关注内容生态、公平性与用户体验之间的动态平衡。它不仅是推荐系统中的安全防线，也是平台治理能力和价值观的重要体现。

4.6 本章小结

本章围绕级联式推荐系统架构中的**召回（Retrieval）**与**过滤限流（Filtering & Throttling）**两大核心模块展开介绍，系统分析了推荐链路最上游阶段的职责定位、技术原理与工程实践。

首先，在召回模块部分，我们从工业级推荐系统面临的海量候选物品检索问题出发，阐述了召回模块存在的必要性，并介绍了多路召回架构的设计思想。在此基础上，对当前主流召回技术进行了分类梳理，包括协同过滤召回、内容与规则召回、Embedding 向量召回以及图召回等典型范式，帮助读者从整体上建立召回技术体系的认知框架。同时，我们进一步介绍了召回结果融合环节的核心流程，包括候选集去重、Quota 配额控制、结果截断、召回兜底等关键机制，说明了工业界如何在召回效果、系统效率与服务稳定性之间实现平衡。

随后，本章介绍了过滤限流模块在推荐系统中的重要作用。作为推荐链路中的安全与治理中枢，过滤限流模块承担着内容质量控制、风险内容拦截以及平台生态治理等职责。围绕这一目标，我们分析了常见的过滤与限流策略，包括违规内容过滤、低质量内容限流、用户反馈驱动的负向调控以及创作者信誉管理等机制，并讨论了过滤限流模块与召回、排序及重排阶段之间的协同关系。

最后，我们结合近年来人工智能技术的发展趋势，介绍了大语言模型（LLM）与视觉语言模型（VLM）在内容安全审核领域的应用。随着 AIGC 内容规模的快速增长，传统依赖规则与专用分类模型的风控体系正逐渐向基于语义理解的统一审核框架演进。LLM 与 VLM 能够从文本、图像、音频及多模态信息中识别更复杂、更隐蔽的风险内容，并与推荐系统深度协同，从而在保障平台安全合规的同时维护良好的用户体验与内容生态。

通过本章的学习，读者应能够理解召回模块如何从海量内容中高效筛选候选集，过滤限流模块如何保障推荐内容的安全与质量，以及二者在整个推荐系统中的协同关系。下一章将进一步进入推荐链路的核心环节：**粗排模块**，重点介绍如何利用机器学习模型对召回候选集进行快速筛选与价值评估，为后续精排阶段提供更加精准的候选集合。

4.7 参考文献

[1] Jinze Bai et al. “Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond”. In: *arXiv preprint arXiv:2308.12966* (2023).

[2] Alexey Dosovitskiy. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[3] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[4] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[5] Junnan Li et al. “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation”. In: *International conference on machine learning*. PMLR. 2022, pp. 12888–12900.

[6] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[7] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.