arXiv:2303.01396v1 [cs.CV] 2 Mar 2023

# MLANet：用于连续视觉与语言导航的带子指令多层级注意力网络

Zongtao He、Liuyi Wang、Shu Li、Qingqing Yan、Chengju Liu 和 Qijun Chen，*IEEE* 高级会员

**摘要—视觉与语言导航（VLN）旨在开发仅通过语言和视觉监督在未见环境中进行导航的智能体。在最近提出的连续设置（连续视觉语言导航）中，智能体必须在自由三维空间中行动，并面临实时执行、复杂指令理解和长动作序列预测等更严峻的挑战。为了在连续视觉语言导航中获得更好的性能，我们设计了一个多层级指令理解流程，并提出了一种新模型——多层级注意力网络（MLANet）。MLANet 的第一步是高效生成子指令。我们设计了快速子指令算法（FSA），将原始指令分割为子指令，并生成一个名为“FSASub”的新子指令数据集。FSA 无需标注，速度比当前方法快 70 倍，因此满足连续视觉语言导航的实时要求。为解决复杂指令理解问题，MLANet 需要对指令和观测进行全局感知。我们提出多层级注意力（MLA）模块，用于融合视觉、低层语义和高层语义，从而生成包含对任务动态且全局理解的特征。MLA 还减轻了噪声词的不利影响，确保对指令的稳健理解。为了在长轨迹中正确预测动作，MLANet 需要关注每一步正在执行的子指令。我们提出峰值注意力损失（PAL），以改进对当前子指令的灵活、自适应选择。PAL 通过使导航智能体将注意力集中于局部信息而发挥作用，从而帮助智能体预测最合适的动作。我们在标准基准上训练并测试 MLANet。实验结果表明，MLANet 以显著优势超越了基线方法。** [1]

索引词—深度学习方法、面向人机交互的多模态感知、面向人机交互的自然对话、基于视觉的导航

## I. 引言

能够遵循自然语言指令的服务机器人一直是机器人研究的长期课题。为实现这一目标，视觉与语言导航（视觉语言导航）[1] 旨在开发一种具身智能体，使其能够在语言监督和视觉反馈的引导下，于非结构化环境中导航。目前，视觉语言导航研究主要在模拟环境中开展，可分为离散和连续两种类型。在 R2R [1]、REVERIE [2] 和 RxR [3]，等离散设置中，智能体只能到达预定义图中的节点，并在节点之间执行理想化的瞬间传送。而在连续设置中 [4]，，智能体可以在连续空间中自由移动，并在每个步骤执行速度控制动作。连续视觉语言导航是一项很有价值的研究方向，因为它模拟了机器人在真实世界中的行为，且不依赖图结构假设。然而，连续视觉语言导航面临实时执行、复杂指令理解和长动作序列预测等挑战，因此需要具备更强的指令利用能力。

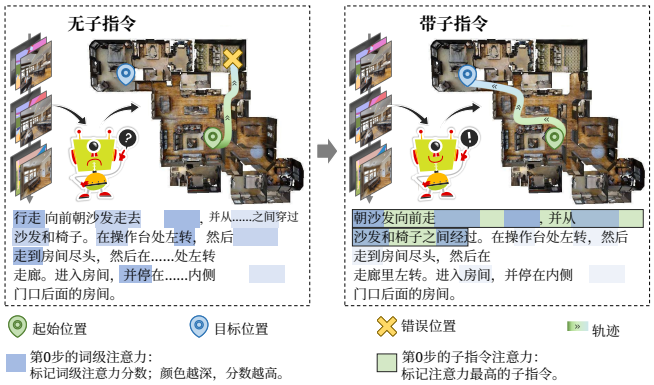



图1. 利用子指令带来的改进。仅使用单词序列注意力（左）时，智能体在初始步骤中误解了任务，最终到达错误位置。子指令（右）强化了有用信息，减弱了噪声词的影响（执行初始步骤时，第三句中的“房间”属于噪声），因此智能体选择了正确方向，并成功导航至目标位置。

基于子指令的多层级指令理解，是解决连续视觉语言导航挑战的一种颇具启发性的方案。设想一个人需要完成一条复杂的导航指令，他/她会怎么做？很自然地，他/她会将指令划分为若干子指令，并逐一完成。子指令的重要性在于，它能以不同的语义粒度提供导航信息，并将复杂任务拆分成易于完成的小任务。在本文中，我们用“层级”表示语言单元的语义粒度。每个单词所包含的信息都非常有限，我们称之为低层语义；而子指令则为导航路径的每个部分提供完整信息，我们称之为高层语义。以往的视觉语言导航方法 [4]–[10] 主要使用特定编码器提取词特征，这些特征包含的语义相对低层；它们既没有将子指令作为整体处理，也没有利用子指令中的高层语义。这种忽视可能导致词语分组问题，即智能体无法正确地将有意义的单词组合在一起。例如，在左侧

本研究得到国家自然科学基金（批准号：61733013、62073245、62173248）以及苏州市重点产业技术创新—核心技术研发计划（编号：SGC2021035）的资助。作者来自机器人与人工智能实验室（RAIL）、同济大学，中国上海 201804。电子邮箱：{xingchen327、wly、lishu、qyan 0131、liuchengju、qjchen}@tongji.edu.cn。[1]代码和数据集可在 https://github.com/ RavenKiller/MLA 获取。

图1中，智能体会注意到一些噪声词（第二句中的“左边”；第三句中的“房间”），因此难以判断该采取什么行动。尽管一些研究 [11]–[13] 已初步尝试利用子指令，但它们仍停留在提取子指令中的词特征，而非高层特征。此外，这些方法并未进一步探索连续视觉语言导航，以发挥子指令的潜力。如果我们同时利用低层语义和高层语义，形成多层级指令理解（如图1右侧所示），引入子指令信息便能在一定程度上减轻噪声词的干扰，并为动作预测提供更精准的指引。

为提升连续视觉语言导航中的性能，我们设计了一套多层级指令理解流程，并提出一种新颖的多层级注意力网络（MLANet）。首先，我们设计了一种高效的无标注算法——快速子指令算法（FSA），将原始指令切分为子指令。以往的子指令算法[11], [12]包含深度神经网络等复杂组件，存在计算效率问题，无法用于实时应用。相比之下，得益于精简的设计，FSA效率很高，可用于数据集预处理和实时导航。我们还公开发布了由FSA处理的数据集“FSASub”，供未来研究使用。随后，我们使用多层级注意力（MLA）模块，对多模态特征进行联合对齐，并融合多层级指令特征。一方面，MLA为智能体提供对导航过程的精确认知，帮助其采取合理行动；另一方面，多模态与多层级特征的融合形成了对指令的全面感知，为智能体提供关于导航路径的丰富上下文。MLA并非仅使用单个子指令中的词特征 [11], [12],，而是利用子指令特征，对复杂指令形成全局理解。最后，我们提出一种名为峰值注意力损失（PAL）的辅助损失函数，用于自适应子指令选择。在导航过程中，智能体可能会考虑错误的或多个子指令，这与我们的预期不同。PAL将注意力分数塑造成单一“峰值”模式，帮助智能体建立对当前子指令的局部感知，使其在长轨迹中能够不受干扰地预测每个动作。借助FSA、MLA和PAL，MLANet降低了子指令生成的时间成本，并同时获得对指令的全局和局部感知，从而克服复杂指令理解和长动作序列预测等挑战。

总而言之，我们的贡献如下：

- 我们提出了一种高效的 FSA，用于生成子指令。该方法无需标注，速度比以往方法快 70 倍。
- 我们提出 MLA，以实现全面的多模态协同接地和多层级语义融合，从而有效对齐指令与观测，并有助于稳健地全局理解复杂指令
- 我们设计了 PAL，用于灵活、自适应地选择子指令。这有助于智能体感知当前子指令的局部信息，并在长轨迹中采取合理的行动。
- 结合 FSA、MLA 和 PAL，我们构建了一种新颖的模型 MLANet。在 VLN-CE 基准上的实验表明，MLANet 的表现显著优于基线方法。我们公开了代码和 FSASub 数据集，以供未来研究使用。

## II. 相关工作

### A. 连续环境中的视觉语言导航

视觉语言导航的最终目的是实现具身导航智能体及其在现实世界中的应用。为实现这一目标，一些研究 [4], [14], [15] 尝试在基于 R2R 的数据集中放宽不切实际的假设，并将视觉语言导航任务置于连续环境中。VLN-CE [4] 是首次尝试为视觉语言导航构建更逼真场景的工作。在 VLN-CE 中，智能体通过单个前向安装的 RGBD 相机感知环境，这与大多数基于 R2R 的研究 [9], [10], [16]–[18]中的全景观测不同。由于状态空间连续、导航步数较长且自我中心视野有限，这项任务更具挑战性。Robo- VLN [14] 通过分层强化学习，尝试解决轨迹长度更长、障碍物更具挑战性的复杂视觉语言导航任务。Li 等人 [19] 利用元学习提升在多目标上的泛化能力，并在 AI2-THOR [15] 中解决视觉无地图导航问题。针对 VLN-CE，已有多种方法 [20]–[22] 被提出用于解决连续视觉语言导航任务。然而，这些方法并未充分关注指令理解，仅利用词特征。我们的工作提出引入子指令，以在连续视觉语言导航中形成多层级指令理解。我们所做的另一项改进是采用 CLIP [23]作为图像编码器，其泛化能力强于以往方法 [24]所采用的 ResNet。

### B. 视觉语言导航中的注意力机制

在视觉语言导航中，利用注意力机制有两种方式。第一种是将软注意力用作跨模态单元。例如，Speaker-Follower [9]和 EnvDrop [10]利用一个或多个软注意力单元，对视觉和指令信息进行协同定位。第二种方式使用预训练的多模态 Transformer [25], [26], ，其由多个自注意力层构成。VLNBERT [17] 使用跨模态 BERT 选择正确的导航路径。Recurrent VLNBERT [27] 在动作解码过程中引入了循环机制。然而，现有方法总是将注意力机制应用于词特征，而忽略了子指令特征。本文提出 MLA，以更加关注多模态和多层级特征的融合。与先前的工作相比，我们使用多头注意力作为 MLA 的基本单元，而不是简单的软注意力，从而增强了对齐不同模态信息的能力。另一个不同之处在于，MLA 实际上利用子指令特征来提供

更稳健的引导，帮助智能体提升对指令的全局感知。因此，MLA 由现有单元构建而成，但针对如何利用导航指令提出了新的思路。

### C. 子指令

现有的视觉语言导航方法很少考虑语言指令的结构特征。经典方法[1], [9]主要使用长短期记忆网络从原始指令中提取词特征，然后在后续处理中应用注意力机制。近期方法 [17], [28]引入 Transformer 来处理语言，但仍然只考虑原始指令。FGR2R [11] 首次尝试理解子指令，每一步只处理一个子指令，并决定是否切换子指令。然而，硬切换策略缺乏对指令的全局感知，且无法从失败步骤中恢复。BabyWalk [13]将已执行的子指令记录到历史缓冲区中，以形成全局感知。SpC-NAV [12] 提出使用指示词来区分和表示子指令。然而，以前的方法 [11]–[13]都停留在处理词特征的层面，并未形成多层级指令理解。此外，这些方法是为离散环境设计的，忽略了子指令生成的时间效率，而这在现实世界应用中是一个关键因素。

本文提出了一套完整的子指令利用流程。首先，我们设计了一种高效的算法 FSA，将原始指令分割为子指令；该算法无需依赖专家标注，并且能够实时处理任何新指令。我们不采用切换策略，而是利用子指令特征，并通过多头注意力动态融合子指令信息，从而确保对指令的全局感知。最后，我们提出 PAL 来塑造注意力分数，促使模型专注于当前子指令，从而有助于局部理解指令。

## III. MLANet

本节介绍我们提出的 MLANet 模型。如图 2 所示，MLANet 包含以下组件：

–**快速子指令算法**，用于高效地为导航智能体提供子指令。– **视觉与语言编码器**，用于将来自不同模态的原始输入嵌入稠密特征空间。– **多层级注意力模块**，用于融合多模态特征和多层级指令特征，以形成对指令的全局理解。– **峰值注意力损失**，用于塑造高层注意力分布，并促进对当前导航进度的局部感知。– **导航推理解码器**，用于接收所有潜在特征并输出预测动作。

本节将阐述问题定义、概述模型架构，并着重介绍 MLANet 中的几个重要组件。

### A. 问题形式化

在视觉语言导航（VLN）中，智能体的目标是在语言监督和视觉反馈的引导下到达目标位置。在导航过程中，智能体会接收到一条完整的指令$I = \{i_1, \ldots, i_L\}$，其中 $i_l$表示第 $l$个词， $L$表示指令的长度。在我们的方法中，智能体还会接收到一个子指令序列 $S = \{s_1, \ldots, s_N\}$，其中 $s_n$ 表示第 $n$个子指令， $N$ 表示子指令的数量。原始指令由 FSA 按照算法1所述切分为子指令。我们的 FSA 具有即插即用的特性，可应用于任何导航模型。

在每个时间步 $t$，智能体会感知一个90°单视角 RGB图像 $o_t^{rgb} \in \mathbb{R}^{256\times256\times3}$ 和深度图像 $o_t^{depth} \in \mathbb{R}^{256\times256}$。智能体使用包含四种动作的动作集合：{向前移动 0.25 米，左转 15°，右转 15°，停止}。令 $\hat{a}_t$ 为真实动作， $a_t$ 为预测动作。执行动作 $a_t$后，智能体的位置从 $p_{t-1}$ 变为 $p_t$，其中 $p_t \in \mathbb{R}^3$ 表示连续环境中的位置向量。

给定起始位置 $P_s$ 和终点位置 $P_e$，当智能体的最终位置 $p_T$满足 $\|P_e - p_T\| \leq \epsilon$时，可判定路径已完成，其中 $T$是智能体预测停止的时间， $\epsilon = 3$米是距离阈值 [1]。总而言之，视觉语言导航任务可以抽象为一个序列到序列问题：智能体接收序列输入 $I$， $S$， $O^{rgb}$， $O^{depth}$，并需输出动作序列 $A = \{a_t\}_{t=1}^{T}$，使智能体能够从起始位置 $P_s$ 移动到终点位置 $P_e$。

### B. 模型架构概述

这里我们介绍 MLANet 的整体架构。如图 2 所示，MLANet 使用编码器-解码器框架。

**编码部分**。视觉与语言编码器用于将原始输入处理为稠密特征，并可利用在大规模数据集上预训练的模型。MLANet 的输入包括深度图像 $O^{depth}$、RGB图像 $O^{rgb}$、指令 $I$，以及来自 FSA 的子指令 $S$ （详见第 III-C 节）。因此，我们使用四个编码器处理不同的模态：深度编码器、RGB编码器、指令编码器和子指令编码器，它们位于图 2 的左侧。

环境观测包括深度图像和RGB图像。对于深度图像 $O^{rgb}$，我们使用ResNet [29]作为深度编码器来提取空间深度特征 $\tilde{v}_t^{depth}$，并使用全连接网络生成低维可学习特征 $v_t^{depth}$。对于RGB图像，我们提出使用CLIP [23] 中的视觉变换器作为RGB编码器，以提取空间RGB特征 $\tilde{v}_t^{rgb}$。经过视觉变换器映射后， [CLS] 位置的特征包含压缩后的视觉信息。因此，我们进一步将 [CLS] 特征输入全连接网络，以生成低维特征 $v_t^{rgb}$。历史信息对导航至关重要，我们使用记忆单元来记忆环境观测，具体内容将在III-D节详细介绍。

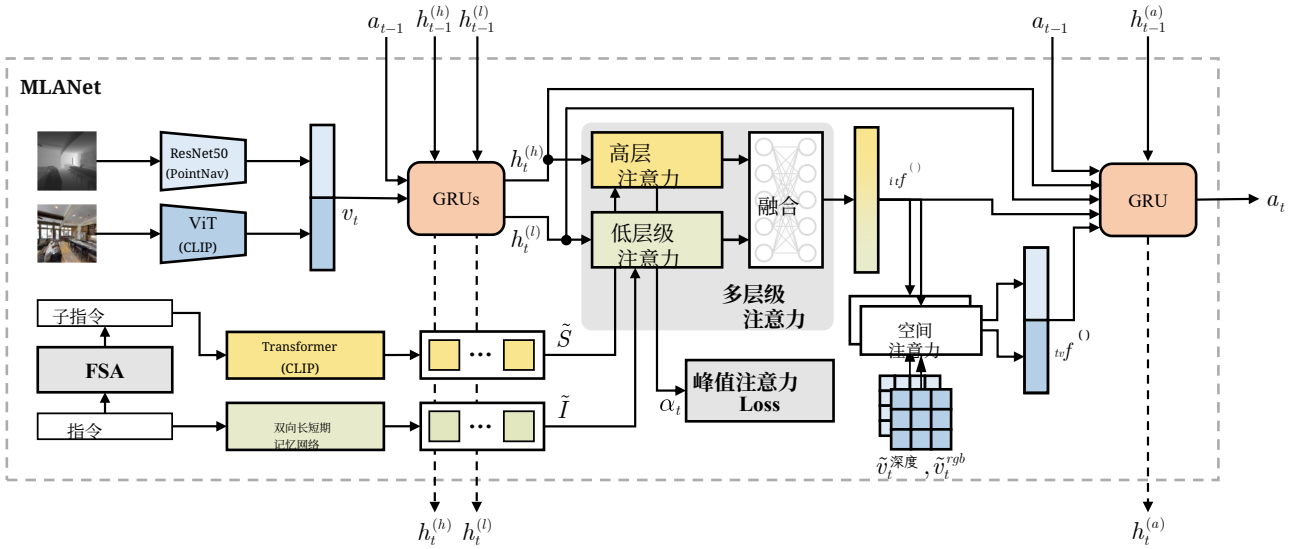



图 2. 模型架构示意图。输入包括一幅 RGB图像、一幅深度图像、一条原始指令，以及来自 FSA 的子指令。经过编码器和 GRU 记忆模块后，MLA 对指令特征进行融合，空间注意力则对视觉特征进行融合。动作解码器接收所有隐藏特征并输出动作。高层注意力的注意力分数将用于 PAL。灰色阴影和粗体字标示了重要组件。

语言模态包含两个层级的语义单元：单词序列和子指令，我们针对它们采用不同的嵌入策略。对于包含单词序列 $I$ 的原始指令，我们使用双向长短期记忆网络作为指令编码器，以$\tilde{}$获取低层指令特征 $I$，并保留对单词含义的追踪。对于子指令，我们使用CLIP [23] 中的文本Transformer作为子指令编码器$\tilde{}$ $\tilde{}$ $\tilde{}$，以获取高层指令特征 $S$。随后，将 $I$ 和 $S$ 投影到低维空间，得到可学习特征。

**注意力部分**。经过编码器后，如何有效聚合不同模态的信息是一个关键问题。我们提出 MLA，动态融合视觉状态 $h_t^{(h)}$，$h_t^{(l)}$ 和多层级指令特征 $\tilde{I}$，$\tilde{S}$，以生成融合特征 $f_t^i$（详见第 III-D 节）。此外，我们提出 PAL，对 MLA 中的高层分数 $\alpha_t$ 进行监督，从而更好地感知指令的局部信息（详见第 III-E 节）。遵循 VLN-CE [4]，，我们进一步使用空间注意力模块，从空间视觉特征中生成融合后的视觉特征 $f_t^v$。

**解码部分**。最后，所有潜在特征都会输入循环动作解码器。动作解码器是一个GRU模块，并生成动作状态 $h_t^{(a)}$，该状态将用于生成预测动作 $a_t$。

### C. 快速子指令算法

原始指令由若干单词组成。在将注意力机制应用于单词特征时，可能会出现词语分组或注意力分散问题。子指令有助于导航过程，因为它们能够提供高层次的指导和易于完成的目标。引入子指令有望减轻噪声词的影响，并克服词语分组问题。现有的子指令分块方法 [11] 包含大量组件，并且耗时较长，无法适用于实时应用。在效率和简洁性这两项原则的指导下：

我们设计了一种新颖的 FSA，将原始指令分割为子指令。算法1展示了 FSA 的简化伪代码。首先，通过若干预定义规则清理原始指令，例如删除转义字符。然后，使用定制的句子分词器对清理后的指令进行粗略分割，该分词器能够查找句子边界，并决定长句子的分割位置。然而，由于导航指令的复杂性，简单的句子分词器在遇到多个连词或歧义表达时可能会失效。因此，我们设计了一组简洁的细化规则，根据词性标注结果细化子指令。该规则集包含若干启发式规则，用于更准确地判断新子指令的起始位置。最后，我们根据词汇表对每个单词进行分词化，从而获得名为“FSASub”的子指令集。FSA 是一种即插即用组件，这意味着除了对数据集进行预处理之外，还可以将 FSA 嵌入导航模型中。图3给出了 FSA 工作方式的定性示例。

### D. 多层级注意力模块

MLA模块的关键功能是融合多模态和多层级特征，如图3所示。多模态特征指来自不同模态的特征，即视觉和语言模态。多层级特征指具有不同语义层级的指令特征。MLA的意义$\tilde{}$在于，它不仅考虑来自单词序列的低层语义 $I$ $\tilde{}$，还利用来自子指令的高层特征 $S$。这种对指令的多层级感知能够有效减轻词级注意力分散所造成的不利影响，并使视觉状态、指令和智能体动作之间形成更好的对齐。经过MLA模块处理后，融合特征 $f_t^{(i)}$ 在时间、空间和语义层面上具备了一定程度的全局感知能力。

**算法1**快速子指令算法

**输入：** 指令集 $D = \{I_i\}_{i=1}^{N}$

**输出：** 子指令集 $O = \{(S_i,$ 词元$_i)\}_{i=}^{N}$ 1

1：初始化 $O \leftarrow \{\}$

2: **对于** $I$ 在 $D$ **执行**

3: 过滤掉 $I$ 中的无效字符，例如 \r\n;

4: 过滤掉 $I$中的无效短语，例如 HTML标记;

5: 使用粗粒度句子分词器将 subs $\leftarrow$ 切分为片段;

6: $S \leftarrow \{\}$;

7: **对于** subs **执行**

8: 单词序列，标签 $\leftarrow$POSTagger(sub);

9: 当前子串 $\leftarrow \emptyset$;

10: $N \leftarrow$单词序列的大小;

11: **对** $i \leftarrow 1$， $N$ **执行**

12: **如果**NeedRefine(i, tags，单词序列) **则**

13: 将当前子串添加到 $S$;

14: **如果**wrods$[i]$ 不是标点符号**则**

15: 当前子串 $\leftarrow$ 单词序列$[i]$;

16: **else**

17: 当前子串 $\leftarrow \emptyset$;

18: **结束条件判断**

19: **else**

20: 当前子串 $\leftarrow$ 当前子串 $+$单词序列$[i]$;

21: **结束条件判断**

22: **结束循环**

23: **如果** 当前子串不等于 $\emptyset$ **则**

24: 将当前子串添加到 $S$中;

25: **结束 if**

26: **结束 for**

27: tokens $\leftarrow$ 对每个元素的单词序列进行分词化;

28: 将（$S$, tokens）添加到 $O$中;

29: **结束 for**

具体而言，MLA的输入包括视觉记忆状态$h_t^{(h)}$， $h_t^{(l)}$、低层指令特征 $\tilde{I}$ 和高层指令特征 $\tilde{S}$。 $h_t^{(h)}$， $h_t^{(l)}$ 来自两个GRU单元，如下所示：

$$h_t^{(h)} = \text{GRU}_h(\text{Concat}(v_t, \tilde{a}_{t-1}); h_{t-1}^{(h)}) \quad (1)$$

$$h_t^{(l)} = \text{GRU}_l(\text{Concat}(v_t, \tilde{a}_{t-1}); h_{t-1}^{(l)}), \quad (2)$$

使用两个记忆单元的原因在于，不同的语义层级关注不同的方面。低层级视觉状态 $h_t^{(l)}$ 倾向于存储与单个单词序列对应的物体或方向信息，而高层视觉状态 $h_t^{(h)}$ 则会考虑来自子指令的长期轨迹信息。随后，我们使用多头注意力而非软注意力，在不同语义层级构建注意力块。低层指令特征 $\tilde{I}$ 和视觉状态 $h_t^{(l)}$ 被输入低层注意力部分，其中 $h_t^{(l)}$ 作为查询，$\tilde{}$ $\tilde{}$和 $I$ 作为键和值。高层指令特征 $S$ 和视觉状态 $h_t^{(h)}$ 被输入高层注意力部分，其中 $h_t^{(l)}$ 作为查询， $\tilde{I}$ 作为键和值。两个输出特征经过拼接后，输入全连接网络，以聚合不同层级的语义：

$$f_t^{(h)} = \text{MultiHead}(h_t^{(h)}, \tilde{S}, \tilde{S}) \quad (3)$$

$$f_t^{(l)} = \text{MultiHead}(h_t^{(l)}, \tilde{I}, \tilde{I}) \quad (4)$$

$$f_t^{(i)} = \text{FC}([f_t^{(h)}; f_t^{(l)}]) \quad (5)$$

其中，MultiHead($Q$， $K$， $V$) 是标准的多头注意力块，[25], FC 是一个全连接层， $f_t^{(i)} \in \mathbb{R}^{512}$是融合指令特征。

$f_t^{(i)}$ 包含哪些信息？MLA 的多层结构有助于注入精确、完整的导航过程信息，为纠正动作提供高层指导。MLA 中的融合层会动态

融合多层级特征，相对增强当前目标并削弱无关信息。此外，多头注意力机制的运用进一步丰富了对多层级的理解。通过这些设计， $f_t^{(i)}$ 不仅能够有效理解“walk”和“kitchen”等单词序列，还能感知“径直穿过房间向前走”等子目标。因此，智能体能够在不同尺度上获得更清晰的全局指令理解，并在导航过程中表现得更好。

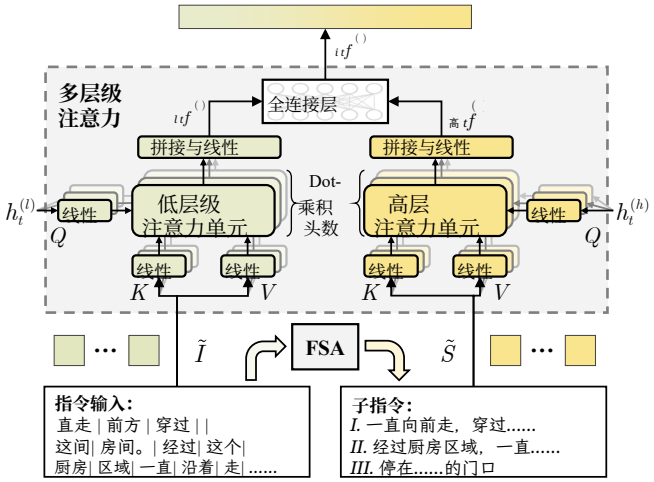



图3. MLA模块的内部结构。两个视觉隐藏状态作为注意力查询输入，多层级指令特征作为键和值输入。经过注意力块后，高层和低层输出通过全连接层融合，生成指令的动态全局感知。

### *E.* 峰值注意力损失

PAL旨在训练过程中塑造高层注意力分数。由于我们将子指令特征用作注意力键，自然会想到高层注意力块如何选择要关注的子指令。我们观察到，MLA模块能够形成对指令的全局感知，但有时也可能导致子指令上的注意力分散问题。也就是说，高层注意力会同时关注多个子指令，而非仅关注一个。注意力分数中出现多个极值点并非我们所期望的，因为智能体可能会对当前导航进度感到困惑。我们希望智能体主要关注单个子指令，以便在动作预测时获得清晰的局部引导。因此，我们提出了一种新的损失函数PAL，以引导模型

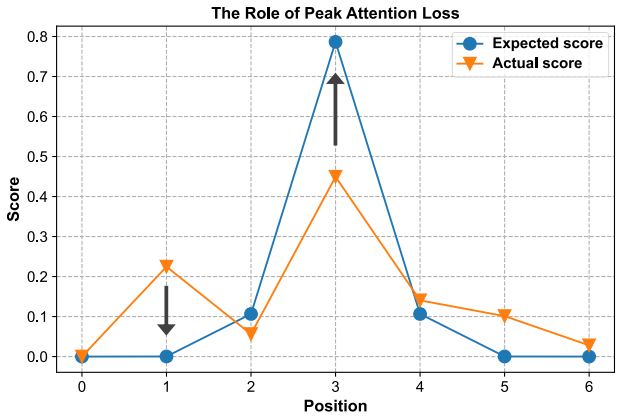



图 4. PAL 的作用。当实际分数（橙色线，三角形标记）中存在多个峰值时，PAL 会强化全局最大点并弱化其他局部最大点，使实际分数更接近期望分数（蓝色线，圆形标记）。

温和地集中关注单个子指令，从而强化对指令的适当局部感知。

首先，从高层注意力中提取注意力分数 $\alpha_t \in \mathbb{R}^N$，并找到最大分数的索引 $k^*$ =Argmax$_k\alpha_{t,\ k}$，其中 $\alpha_{t,\ k}$ 表示向量 $\alpha_t$ 中的第 $k$ 个元素。然后，构建期望分数向量$\beta_t \in \mathbb{R}^N$。所提出的损失函数通过 $\beta_t$ 与 $\alpha_t$之间的均方误差计算：

$$z_k = -\frac{(k-k^*)^2}{2\sigma^2}, k = 1, \ldots, N \tag{6}$$

$$\beta_{t,k} = \frac{e^{z_k}}{\sum_{j=1}^{N} e^{z_j}} \tag{7}$$

$$\mathcal{L}_{peak} = \frac{1}{N}\sum_{t=1}^{T}\sum_{k=1}^{N}(\alpha_{t,k} - \beta_{t,k})^2 \tag{8}$$

$\beta_t$ 中的元素构成一组类似于高斯曲线的序列，而 $\sigma$是一个用于控制曲线形状的超参数（称为 PAL 聚焦比例）。在式（6）和式（7）中，我们利用类高斯曲线构建期望分布。不过，也可以选择具有“单峰”形状的各种曲线类型。我们将在实验 IV-G 中讨论线性、二次和三次下降曲线等不同曲线类型的影响。另一个值得注意之处是，我们不会将梯度传递给 Argmax 算子，以确保可微性。

图 4 直观展示了 PAL 的工作原理。当实际注意力分数 $\alpha_t$ 存在多个局部最大值时，PAL 会将其分布调整为预期分数 $\beta_t$，使其只有一个全局最大值。同时，PAL 允许最大值周围的区域有少量信息泄漏。这种泄漏确保智能体在决定是否转移注意力时拥有必要的上下文。总的来说，PAL 有助于 MLA 模块的训练，并促进子指令与当前导航进度之间的对齐。

### *F.* 动作解码器

动作解码器是一个 GRU 模块，它接收所有潜在特征并输出预测动作：

$$h_t^{(a)} = \text{GRU}_a(\text{Concat}(h_t^{(h)}, h_t^{(l)}, \tilde{a}_{t-1}, f_t^{(i)}, f_t^{(v)}); h_{t-1}^{(a)}) \tag{9}$$

$$\bar{a}_t = \text{Softmax}(\text{Linear}(h_t^{(a)})) \tag{10}$$

$$a_t = \text{Argmax}_k(\bar{a}_{t,k}) \tag{11}$$

动作分布 $a_t$ 将用于构造动作损失 $\mathcal{L}_{action}$ = 交叉熵($a_t$, $\hat{a}_t$)，其中 $\hat{a}_t$ 为真实标签。遵循 Ma 等人 [16] 和 Zhu 等人 [30], 的做法，我们还采用进度监控损失 $\mathcal{L}_{progress}$。结合这两种损失与 PAL，我们得到总训练损失如下：

$$\mathcal{L} = \mathcal{L}_{action} + \lambda\mathcal{L}_{peak} + \theta\mathcal{L}_{progress} \tag{12}$$

$\lambda$ 和 $\theta$ 是用于平衡不同组成部分的参数。通过网格搜索，我们经验性地选择 $\lambda = 0.4$ 和 $\theta = 1.0$ 作为最佳设置。对于 $\lambda$，我们在训练过程中采用线性递增策略（从 0 增至 0.4）来控制该参数，以避免约束过强。

## IV. 实验

### *A.* 实验设置

*1*）数据集：我们使用 VLN-CE 数据集及提出的子指令集 FSASub 来训练和评估所提出的 MLANet 模型。VLN-CE [4] 是 Habitat模拟器 [31],中 R2R[1]数据集的重建版本，共包含 4475 条连续轨迹和 13425 条对应指令。在模拟器中，智能体配备了一台自我中心RGBD相机，分辨率为 $256 \times 256$（RGB图像下采样至 $224 \times 224$ 以适配编码器），水平视场角为 90°。VLN-CE 中的所有样本分为四个子集：训练集、已见验证集、未见验证集和测试集。遵循 VLN-CE 的设定，智能体可执行四种类型的动作：向前移动 0.25 米、右转 15°、左转 15°或停止。然而，原始 VLN-CE 数据集不包含子指令。我们使用第 III 节提出的 FSA，获得一个包含子指令的数据集，称为“FSASub”。为证明 FSA 的有效性，我们还重新实现了 FGR2R [11]中的分块方法，并用它生成一个称为“FGSub”的数据集。后续章节将展示对比实验。

*2*）评估指标：我们采用以往工作中的五项指标 [1], [4] 来评估模型的性能：

- **TL**：轨迹长度，指实际路径的总长度。
- **NE**：导航误差，指停止位置与目标位置之间的距离。
- **nDTW**：归一化动态时间规整，用于衡量两个时间序列之间的相似度。
- **OSR**：预言机成功率，即采用预言机停止规则时的成功率。
- **SR**：成功率。当智能体距离目标不超过 3 米时，该轨迹判定为成功。
- **SPL**：成功率加权逆路径长度，是成功率除以轨迹长度。遵循

表 I 验证集上的实验结果及所有指标。DA 和 AUG 分别表示 DAgger 训练和 EnvDrop 增强训练。已见验证集包含与训练集相同的场景，而未见验证集包含新的未见场景。粗体字表示某项指标的最佳值。 †：基线方法未采用强化学习训练，因此我们在相同条件下微调官方预训练权重。

| | | | | | 已见验证集 | | | | | | 未见验证集 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | 模型 | DA | AUG | RL | TL↓ | NE↓ | nDTW↑ | OSR↑ | SR↑ | SPL↑ | TL↓ | NE↓ | nDTW↑ | OSR↑ | SR↑ | SPL↑ |
| 1 | Seq2Seq† [4] | - | - | - | 8.34 | 8.48 | 0.47 | 0.32 | 0.22 | 0.21 | 8.93 | 9.28 | 0.40 | 0.28 | 0.17 | 0.15 |
| 2 | | ✓ | ✓ | - | 9.37 | 7.02 | 0.54 | 0.46 | 0.33 | 0.31 | 9.32 | 7.77 | 0.47 | 0.37 | 0.25 | 0.22 |
| 3 | | ✓ | ✓ | ✓ | **7.30** | 7.55 | 0.51 | 0.38 | 0.30 | 0.28 | **6.19** | 7.97 | 0.48 | 0.26 | 0.21 | 0.20 |
| 4 | CMA† [4] | - | - | - | 8.51 | 8.17 | 0.47 | 0.35 | 0.28 | 0.26 | 7.87 | 8.72 | 0.44 | 0.28 | 0.21 | 0.19 |
| 5 | | - | ✓ | - | 8.49 | 8.29 | 0.47 | 0.36 | 0.27 | 0.25 | 7.68 | 8.42 | 0.46 | 0.30 | 0.24 | 0.22 |
| 6 | | ✓ | ✓ | - | 9.06 | 7.21 | - | 0.44 | 0.34 | 0.32 | 8.27 | 7.60 | - | 0.36 | 0.29 | 0.27 |
| 7 | | ✓ | ✓ | ✓ | 8.22 | 7.63 | 0.51 | 0.34 | 0.25 | 0.23 | 7.70 | 7.77 | 0.49 | 0.32 | 0.25 | 0.23 |
| 8 | MLANet（我们的方法） | - | - | - | 8.20 | 7.56 | 0.51 | 0.37 | 0.30 | 0.28 | 7.64 | 8.32 | 0.46 | 0.29 | 0.23 | 0.21 |
| 9 | | - | ✓ | - | 8.49 | 7.10 | 0.55 | 0.44 | 0.35 | 0.33 | 7.51 | 7.98 | 0.48 | 0.30 | 0.24 | 0.23 |
| 10 | | ✓ | - | - | 9.70 | 7.10 | 0.52 | 0.46 | 0.33 | 0.30 | 9.53 | 7.67 | 0.48 | 0.39 | 0.28 | 0.26 |
| 11 | | ✓ | ✓ | - | 9.07 | 6.04 | 0.58 | **0.52** | 0.42 | 0.39 | 8.64 | 7.24 | 0.52 | 0.40 | 0.32 | 0.30 |
| 12 | | ✓ | ✓ | ✓ | 8.10 | **5.83** | **0.60** | 0.50 | **0.44** | **0.42** | 7.21 | **6.30** | **0.58** | **0.42** | **0.38** | **0.35** |

以往的研究 [1], [4], [32], ，我们使用 SPL 选择最佳模型。

*3*）训练细节： 我们使用 CLIP 的 ViT-B/32 版本作为 RGB图像编码器和子指令编码器。投影后的 RGB特征、深度特征、词特征和子指令特征的维度均为 256。模型中的所有 GRU 隐藏状态和融合特征均为 512 维。为缓解过拟合问题并增强模型的泛化能力，我们以 0.25 的比例对特征进行随机丢弃。基础模型通过模仿学习进行训练，即模型尝试模仿数据集中的专家动作。我们使用 Ada m 作为训练优化器，将批量大小设置为 5，学习率设置为 2.5e-4。此外，我们使用 3.2 作为转折加权系数 [33] ，以鼓励智能体做出更多样化的决策。由于 PAL 是一种较强的约束，可能会在训练早期引入噪声，因此我们对超参数 $\lambda$采用线性递增策略。我们训练模型 45 个训练轮次，每 5 个训练轮次评估一次模型，并在未见验证集划分上选择最佳模型进行分析，与先前工作 [1], [4]保持一致。除基础模型外，我们还进一步使用数据集聚合（ DAgger） [34], 说话者增强 [10]和近端策略优化 [35] 训练最佳模型，以与当前方法进行比较。最佳模型采用增强训练（动作损失、进度损失和 PAL）以及 DAgger 训练（动作损失和 PAL）进行训练，并使用 PPO 进行微调。我们在多个 GPU服务器上训练模型，以减轻设备偏差，包括 GeForce RTX 3090、GeForce RTX 2080 和 Tesla T4。更多训练细节请参见代码仓库中的配置文件。

### *B.* 与基线方法的比较

表 I 展示了我们提出的 MLANet 与基线方法 [4]的性能对比。Seq2Seq [4] 模型非常简单，不具备动态指令理解能力。CMA [4] 模型采用软注意力来实现动态指令理解。为进行公平比较，我们还采用了 Seq2Seq 和 CMA 模型所使用的基于RNN的结构作为我们的框架。实验结果表明，我们提出的方法是有效的，且性能优于先前的方法。辅助方法 DA 和 AUG 分别表示 DAgger 训练和 EnvDrop 增强训练。RL 表示通过少量 PPO 步骤对模型进行微调。遗憾的是，基线方法没有公开其预训练权重，因此我们无法在 RL 设置下对其进行微调。

*1)* 不使用辅助训练方法：我们发现，仅使用模仿学习进行训练时，MLANet 的表现优于 Seq2Seq 和 CMA。与 Seq2Seq 相比（第 1 行与第 8 行），MLANet 在已见验证集上的 SPL 从 0.21 提升至 0.28（相对提升 33%），在未见验证集上的 SPL 从 0.15 提升至 0.21（相对提升 40%）。与 CMA 相比（第 4 行与第 8 行），MLANet 在已见验证集上的 SPL 从 0.26 提升至 0.28（相对提升 8%），在未见验证集上的 SPL 从 0.19 提升至 0.21（相对提升 10%）。在其他指标上，MLANet 模型也有所提升。

*2)* 使用辅助训练方法：我们还使用辅助训练方法对 MLANet 进行训练。AUG 训练（第 9 行与第 8 行）使 MLANet 在未见验证集上达到 0.23 SPL，提升了 0.02 S PL（相对提升 10%）。DA 训练（第 10 行与第 8 行）使 MLANet 在未见验证集上达到 0.26 SPL，提升了 0.05 S PL（相对提升 24%）。AUG 和 DA 的效果显然有所不同。AUG 大幅提升了已见验证集上的成功率，但泛化能力不足；DA 大幅提升了未见验证集上的成功率，但在已见验证集上的表现不如 AUG。原因在于，AUG 训练包含许多来自已见场景的样本，有助于在已见验证集上进行导航；而 DA 训练则鼓励探索，并提升模型在未见环境中的泛化表现。同时应用这两种辅助方法时，MLANet 模型的表现显著提升，在已见验证集上达到 0.39 SPL，在未见验证集上达到 0.30 SPL。经过强化学习微调后，性能进一步明显提升，在未见验证集上达到 0.35 SPL。

在相同的训练集-

设置下，MLANet 也优于基线方法。首先，MLANet (AUG) 在已见验证集和未见验证集上的 SPL 均优于 Seq2Seq (DA, AUG)（第 9 行与第 2 行相比）。其次，MLANet (AUG) 在已见验证集和未见验证集上的 SPL 均优于 CMA (AUG)（第 9 行与第 5 行相比）。第三，MLANet (DA, AUG) 在已见验证集和未见验证集上的 SPL 均更高（第 11 行与第 2、6 行相比）。最后，经过强化学习微调后，MLANet (DA,AUG, RL) 以显著优势超越基线方法（第 12 行与第 3、7 行相比）。

一个令人惊讶的现象是，与 MLANet 的性能提升不同，Seq2Seq 和 CMA 在经过强化学习微调后，表现反而变差。尽管 Seq2Seq（DA、AUG、RL）的轨迹长度最短，但其较低的成功率表明，轨迹较短是因为智能体在到达目标之前错误地停止了。这一现象表明，MLANet 比基线方法更具潜力，而更强大的训练方法有望释放这种潜力。

总而言之，实验结果表明，与基线方法相比，所提出的模型在几乎所有指标上都取得了更好的表现，证明了我们方法的有效性。

### *C.* 排行榜上的表现

按照 VLN-CE 挑战赛的要求，我们将最佳模型的结果提交至官方测试排行榜。表 II 展示了与基线方法及其他几种方法的性能比较。MLANet 模型在几乎所有指标上都优于 CM2 [20],CMA [4], LAW[21],WPN+DN [22] 和 HPN+DN [22] 。与 CMA 基线相比，MLANet 有助于智能体更好地理解指令和导航目标，从而采取更恰当的行动来完成视觉语言导航任务。HPN+DN [22] 是一种分层决策模型，需要进行数据集修改和大量的强化学习迭代。与 HPN+DN 相比，MLANet 的 SPL 也有所提升。MLANet 唯一未取得最佳成绩的指标是导航误差（NE），其数值相对高于 HPN+DN。我们认为，这是因为 HPN+DN 在训练过程中利用了图优先级，因此比不使用离散图信息的 MLANet 更接近目标。

表 II 测试排行榜对比。

| | Test | | | | |
|---|---|---|---|---|---|
| **模型** | **TL↓** | **NE↓** | **OSR↑** | **SR↑** | **SPL↑** |
| CM2 [20] | 13.85 | 7.74 | 0.39 | 0.31 | 0.24 |
| CMA [4] | 8.85 | 7.91 | 0.36 | 0.28 | 0.25 |
| LAW [21] | 9.67 | 7.69 | 0.38 | 0.28 | 0.25 |
| WPN+DN [22] | 9.68 | 7.49 | 0.36 | 0.29 | 0.25 |
| HPN+DN [22] | 8.02 | **6.65** | 0.37 | 0.32 | 0.30 |
| MLANet（我们的方法） | **7.42** | 6.78 | **0.39** | **0.34** | **0.32** |

### *D.* 模块消融研究

为研究不同组件的作用，我们在表III中分别对 FSA、MLA 和 PAL 进行了消融。辅助训练在时间和存储成本方面都非常耗费资源，因此，为提高时间效率，我们仅通过简单的模仿学习开展所有消融实验。

移除 FSA 时，会屏蔽所有高层指令特征，因此在保持其他结构不变的同时，不再提供子指令信息。实验表明，在没有 FSA 提供子指令的情况下，模型在两个验证集上的表现都很差。

移除 MLA 时，会对指令特征取平均并将其拼接在一起，因此不再使用多层级注意力。采用简单平均策略处理指令特征时，SR 和 SPL 会下降，因为模型无法清晰地感知指令。这一结果进一步证明，指令的多层次理解对我们的模型至关重要。

移除 PAL 时，平衡参数 $\lambda$ 设为零。在我们的方法中，PAL 是训练 MLA模块的有效工具。移除 PAL 后，SPL 下降，这表明 PAL 有助于训练出更好的模型。

总而言之，由于多层级指令理解流程中缺少任何一个部分，模型性能都会下降。得益于 FSA、MLA 和 PAL，完整模型能够细致地理解指令，并具备出色的能力，能够以相对较短的路径到达正确位置。

### *E. FSA* 的性能

在多项工作中，FGR2R [11] 与我们的方法设置最为接近 [6], [12], [13], ，因此我们选择它作为比较对象。为研究不同子指令分割方法的效果，我们生成了两个数据集 FGSub（来自 FGR2R [11]）和 FSASub（来自 FSA），然后分别用它们训练模型。在训练过程中，我们使用超参数 $\theta = 1.0$, $\lambda = 0.4$, $\sigma = 0.6$。表 IV 比较了这些子指令集的不同方面，其中时间是处理整个数据集所需的时间；分割比例是成功分割的指令所占的比例；平均数量是每条指令中的子指令平均数量。结果表明，在已见验证集和未见验证集上，使用 FSASub 训练的模型表现都略优于 FGSub。原因在于，FSASub 能够成功分割更多指令，并提供更多子指令用于训练。更高的分割比例和 FSASub 的子指令平均数量证明了这一优势。图 5 所示的核密度估计（KDE）图也表明，与 FGSub 相比，FSASub 的子指令数量分布更加平滑，因此更有利于训练。FSA 的另一项优势是时间效率。生成 FGSub（约 8.8 小时）所需的时间，是生成 FSASub（仅 7.3 分钟）所需时间的 72 倍以上。

下面给出一个定性示例。对于同一条指令，两种分段方法得出了不同的结果。FGR2R 方法未能检测出由“and”连接的最后两个子指令，而我们的方法成功地将其分段。这是因为我们在 FSA 中采用的细化规则能够更精确地检测子指令的起始边界。

**FGSub**

表III 模块消融研究。

| 模型 | 已见验证集 | | | | | | 未见验证集 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TL↓ | NE↓ | nDTW↑ | OSR↑ | SR↑ | SPL↑ | TL↓ | NE↓ | nDTW↑ | OSR↑ | SR↑ | SPL↑ |
| MLANet | **8.20** | **7.56** | **0.51** | 0.37 | **0.30** | **0.28** | **7.64** | **8.32** | **0.46** | 0.29 | **0.23** | **0.21** |
| 不含 FSA | 8.62 | 7.93 | 0.48 | 0.37 | 0.28 | 0.26 | 8.20 | 8.32 | 0.45 | **0.30** | 0.21 | 0.19 |
| 不含 MLA | 8.93 | 8.13 | 0.48 | **0.39** | 0.28 | 0.26 | 8.47 | 8.49 | 0.43 | 0.30 | 0.21 | 0.19 |
| 不含 PAL | 8.67 | 7.62 | 0.50 | 0.39 | 0.29 | 0.27 | 8.02 | 8.77 | 0.43 | 0.28 | 0.20 | 0.18 |

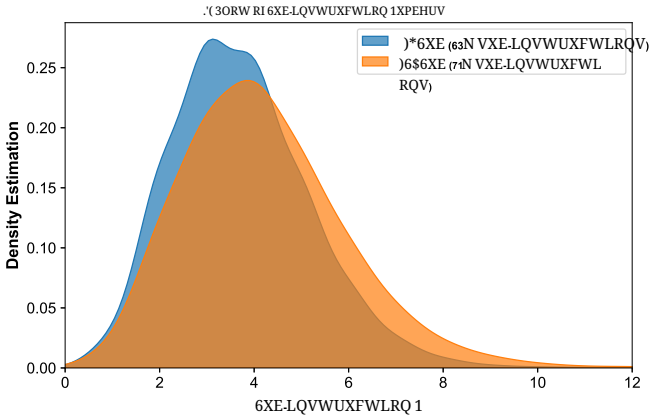



图 5. 子指令数量的核密度估计（KDE）图。子指令数量越多，表示一条指令中包含的子指令越多。

表 IV 两个子指令集的比较。SR 和 SPL 是使用不同子指令集训练的模型性能。

| Name | 已见验证集 | | 未见验证集 | | 片段比例 | Avg Num | Time（小时） |
|---|---|---|---|---|---|---|---|
| | SR | SPL | SR | SPL | | | |
| FGSub | 0.294 | 0.280 | 0.219 | 0.205 | 97.56% | 3.72 | 8.80 |
| FSASub | **0.301** | **0.284** | **0.228** | **0.214** | **98.36%** | **4.20** | **0.12** |

- 向右转
- 经过冰箱
- 左转，走到靠近入口和餐厅区域的走廊处

**FSASub**

- 向右转
- 经过冰箱
- 左转
- 然后走到靠近入口和餐厅区域的走廊处

总而言之，FSASub 在三个方面优于 FGSub。首先，FSASub 能为模型训练提供更有效的子指令。其次，生成 FSASub 比生成 FGSub 更省时。最后，使用 FSASub 训练的模型表现优于使用 FGSub 训练的模型。因此，我们默认使用 FSASub 作为子指令集来训练模型。

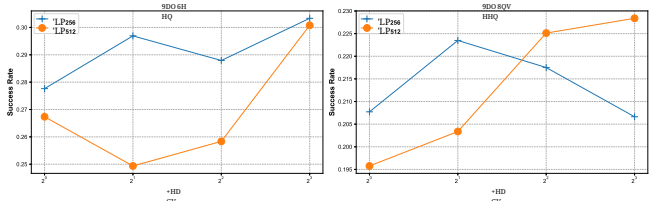



图6. MLA中不同超参数设置下已见验证集（左）和未见验证集（右）的成功率曲线。维度256表示隐藏维度为256；维度512表示隐藏维度为512。

### *F. MLA*的超参数

注意力头数和隐藏维度是 MLA 中两个关键的超参数。我们在图6中研究了它们的影响。总体而言，维度256在已见验证集上表现良好，但在未见验证集上的表现不如维度512。这些结果表明，隐藏维度更大的模型由于具有更强的学习潜力，在未见环境中泛化能力更好。至于头数，结果则相对复杂。直观来看，更多的头数能够更细致地理解输入，因此应带来更好的性能。这一假设在维度512时成立：随着头数增加，成功率曲线持续上升。然而，当隐藏维度设为256时，未见验证集上的成功率在头数为2时达到最高点，随后开始下降。在多头注意力单元中，每个头都有不同的关注重点。如果总维度过低而头数过多，所有头的表征能力都会不足。这就是维度256在未见验证集上的表现无法持续提升的原因。我们希望模型能够在未见环境中实现良好的泛化，因此选择维度512、8个头作为 MLA 的配置。

### *G. PAL*的超参数

曲线类型和聚焦比率 $\sigma$ 是 PAL 的两个重要超参数。我们在表V中探究了它们的影响。我们比较了五种曲线类型：高斯、常数、线性、二次和三次，并将在本节中使用相应的曲线类型命名在每种设置下训练的模型。高斯曲线由公式（6、7、8）定义。后四种类型是次数不同的多项式，用于控制从峰值到边界的下降趋势。我们发现，高斯曲线取得了最佳性能，而二次曲线在未见验证集上具有具有竞争力的性能。然而，二次曲线在已见验证集上的表现并不理想，显示出较少

对更好的导航智能体的帮助。对于我们使用的高斯曲线，$\sigma$ 控制注意力分数的预期形状。较小的 $\sigma$ 会使注意力分数形成更尖锐的峰值。在改变 $\sigma$时，我们发现了一个有趣的现象：使用 $\sigma = 0.8$ 的模型在已见验证集上取得了最高成功率。然而，它在未见验证集上的表现较差，已见验证集与未见验证集之间的 SPL 相差 0.11。最后，我们决定使用带有 $\sigma = 0.6$的高斯曲线。这种组合在训练中平衡了硬注意力和软注意力策略，有助于模型取得良好性能。

采用不同曲线类型（TYPE）和 PAL 聚焦比例（$\sigma$）时的模型性能。

| 模型 | TYPE | $\sigma$ | 已见验证集 SR | 已见验证集 SPL | 未见验证集 SR | 未见验证集 SPL |
|---|---|---|---|---|---|---|
| MLANet（使用 PAL） | 高斯 | 0.5 | 0.275 | 0.257 | 0.209 | 0.194 |
| | | 0.6 | 0.301 | 0.284 | **0.228** | **0.214** |
| | | 0.8 | **0.312** | **0.295** | 0.198 | 0.185 |
| | | 1.0 | 0.289 | 0.268 | 0.214 | 0.198 |
| MLANet（采用 PAL） | 常数 | - | 0.276 | 0.260 | 0.187 | 0.174 |
| | 线性 | - | 0.263 | 0.250 | 0.209 | 0.196 |
| | 二次 | - | 0.240 | 0.226 | 0.226 | 0.212 |
| | 三次 | - | 0.261 | 0.244 | 0.210 | 0.196 |
| | 高斯 | 0.6 | **0.301** | **0.284** | **0.228** | **0.214** |

### *H.* 可视化

图7展示了我们的方法完成的一条路径。上半部分是若干步骤的 RGB图像，下半部分是随时间步变化的注意力图。

在最初的几个步骤中，智能体专注于子指令“直穿房间向前走”。然而，它身处阳台上，看不到任何房间。因此，智能体不断左转以寻找房间。看到房间场景后，智能体意识到它应该穿过这个房间并向前移动。随后，注意力峰值逐渐转向第二条子指令“经过厨房区域，一直走到正前方的门口”。在第13步，智能体专注于第二条子指令，并自信地向前移动。“厨房”是路径第二部分的关键词，因此智能体在第24步左转，确认自己是否仍在厨房区域。在第36步，智能体逐渐决定关注最后一条子指令“停在卧室门口”，并向前移动到达门口。最后，智能体在第51步停止，站在门旁并看到了卧室。注意力热图呈对角线分布，这意味着模型成功学会了随时间转移注意力。得益于 PAL，几乎每一步都只有一条子指令受到关注。

为了全面分析我们的方法，我们还展示了两个失败案例。图8给出了两个不理想的注意力分布
智能体未能到达目标。其中一种情况（图8上半部分）出现了振荡，因为指令过于简单，且包含一个罕见的物体——“点亮的枝形吊灯”。另一种情况（图8下半部分）在整个路径中只关注一个子指令。检查对应路径后，我们发现路径主要位于一条很长的走廊中，因此模型始终关注子指令“然后进入房子”。这些失败案例表明，我们的方法容易受到罕见语言或视觉观察的影响，错误的注意力模式会损害对导航目标的理解，并误导动作预测。如何减轻这种现象是一个值得未来研究的课题。

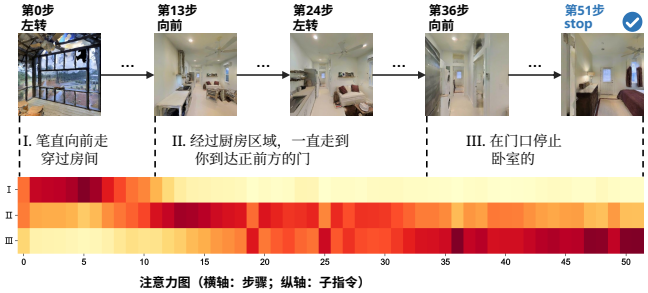



图7. MLANet模型的一个定性示例。上半部分展示视觉捷径，下半部分展示子指令注意力（MLA中的高层注意力）。

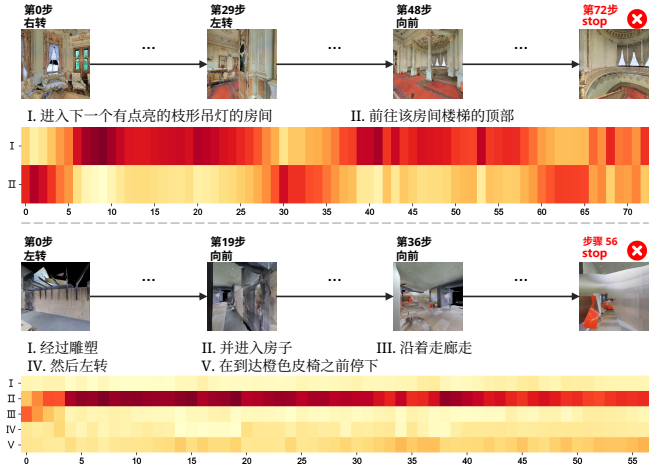



图8. 子指令注意力图的失败案例。

## V. 结论

本文介绍了一种多层级指令理解流程，并提出 MLANet，以提升连续视觉语言导航中的导航性能。多层级指令理解流程包含子指令分段、多层级语义特征融合和自适应子指令选择。我们首先设计了一种快速子指令算法 FSA，用于生成子指令，并获得名为“FSASub”的子指令集。FSA 是一种无标注算法，效率显著高于现有方法。随后，我们提出 MLA 模块，用于融合不同语义层级的特征，并根据子指令提供精准的导航指导。得益于 MLA，智能体能够动态地全局感知指令。最后，我们提出一种新的损失函数 PAL，以无监督方式调整注意力分数

并帮助智能体选择当前子指令，从而确保其能够局部感知当前目标。实验表明，带有 MLA 模块的模型表现优于基线方法，在已见验证集、未见验证集和测试集上的 SPL 分别提升了 31%、30% 和 28%。消融研究和超参数分析进一步详细说明了我们的方法。

这项工作仍存在一些局限。MLANet采用了简单的基于RNN的框架，这可能会限制对子指令和观测的充分利用。此外，MLANet有时会对罕见的观测较为敏感。基于Transformer的框架或许是我们改进特征表示的下一步选择。未来的研究将着重引入更稳健的模型瓶颈，并探索更丰富的技术来处理子指令。我们还计划在其他视觉语言导航或面向目标的视觉-语言导航数据集上，尝试子指令思维方法。

## 参考文献


[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. S¨underhauf, I. Reid, S. Gould 和 A. van den Hengel，“视觉与语言导航：在真实环境中解读以视觉为依据的导航指令”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2018年，第3674–3683页。[2] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen 和 A. van den Hengel，“REVERIE：真实室内环境中的远程具身视觉指代表达”，载于*IEEE/CVF*计算机视觉与模式识别会议论文集，2020年，第9979–9988页。[3] A. Ku, P. Anderson, R. Patel, E. Ie 和 J. Baldridge，“跨房间：具有密集时空定位的多语言视觉与语言导航”，载于自然语言处理实证方法会议论文集，2020年，第4392–4412页。[4] J. Krantz, E. Wijmans, A. Majumdar, D. Batra 和 S. Lee，“超越导航图：连续环境中的视觉与语言导航”，载于欧洲计算机视觉会议论文集。Springer，2020年，第104–120页。[5] X. Wang, W. Xiong, H. Wang 和 W. Y. Wang，“三思而后行：弥合无模型与基于模型的强化学习，以实现前瞻式视觉与语言导航”，载于欧洲计算机视觉会议论文集（*ECCV*），2018年，第37–53页。[6] Y. Hong, C. Rodriguez-Opazo, Y. Qi, Q. Wu 和 S. Gould，“用于智能体导航的语言与视觉实体关系图”，载于神经信息处理系统会议，2020 [ ] 2010 09304 。在线。获取地址：[7] http://arxiv.org/abs/。T.-C. Chi, M. Shen, M. Eric, S. Kim 和 D. Hakkani-tur，“直接提问：一种用于视觉与语言导航的交互式学习框架”，*AAAI*人工智能会议论文集，第34卷，第03期，第2459–2466页，2020年。[8] Q. Xia, X. Li, C. Li, Y. Bisk, Z. Sui, J. Gao, Y. Choi 和 N. A. Smith，“视觉与语言导航的多视角学习”，*arXiv*预印本 *arXiv:2003.00857*，2020年。[9] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein 和 T. Darrell，“用于视觉与语言导航的说话者-跟随者模型”，载于神经信息处理系统会议论文集，2018年，第3318–3329页。[10] H. Tan, L. Yu 和 M. Bansal，“学习在未见环境中导航：结合环境丢弃的回译”，载于*NAACL-HLT*论文集，第*1*卷（长篇与短篇论文），2019年，第2610–2621页。[11] Y. Hong, C. Rodriguez, Q. Wu 和 S. Gould，“子指令感知视觉-语言导航”，载于自然语言处理实证方法会议论文集，2020年，第3360–3376页。[12] Y. Zhang, Q. Guo 和 P. Kordjamshidi，“通过对空间配置进行推理实现导航”，载于第二届空间语言理解与机器人落地交流国际联合研讨会论文集，2021年，第42–52页。[13] W. Zhu, H. Hu, J. Chen, Z. Deng, V. Jain, E. Ie 和 F. Sha，“BabyWalk：通过迈出小步，在视觉与语言导航中走得更远”，载于计算语言学协会第*58*届年会论文集。在线：计算语言学协会，2020年7月，第2539–2556页。 [在线]。获取地址：https://aclanthology.org/2020 229 .acl-main。M. Z. [14] Irshad, C.-Y. Ma 和 Z. Kira，“用于机器人视觉与语言导航的分层跨模态智能体”，*arXiv*预印本 *arXiv:2104.10674*，2021年。

[15] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta 和 A. Farhadi，“AI2-THOR：用于视觉人工智能的交互式三维环境”，arXiv预印本 arXiv:1712.05474，2017年。[16] C.-Y. Ma, J. Lu, Z. Wu, G. AlRegib, Z. Kira, R. Socher 和 C. Xiong，“通过辅助进度估计实现自我监控导航智能体”，ICLR论文集，2019年。[17] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh 和 D. Batra，“利用来自网络的图像-文本对改进视觉与语言导航”，载于欧洲计算机视觉会议论文集。Springer，2020年，第259–274页。[18] Z. Zhan, L. Lin 和 G. Tan，“面向远程具身视觉指代表达的物体感知导航”，《神经计算》，第515卷，第68–78页，2023年。 [在线]。获取地址：https://www.sciencedirect.com/science/article/pii/S0925231222012942 [19] F. Li, C. Guo, B. Luo 和 H. Zhang，“利用元学习和场景先验实现室内多目标、多场景视觉无地图导航”，《神经计算》，第449卷，第368–377页，2021年。[在线]。获取地址：https://www.sciencedirect.com/science/article/pii/S0925231221004707[20] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth 和 K. Daniilidis，“用于视觉与语言导航的跨模态地图学习”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2022年6月，第15 460–15 470页。[21] S. Raychaudhuri, S. Wani, S. Patel, U. Jain 和 A. Chang，“用于连续环境中视觉与语言导航的语言对齐航点（LAW）监督”，载于自然语言处理实证方法会议论文集，2021年，第4018–4028页。[22] J. Krantz, A. Gokaslan, D. Batra, S. Lee 和 O. Maksymets，“ 用于指令引导导航的连续环境航点模型 ”，载于IEEE/CVF ICCV论文集，2021年，第15 162–15 171页。[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark 等，“通过自然语言监督学习可迁移的视觉模型”，arXiv预印本 arXiv:2103.00020，2021年。[24] K. He, X. Zhang, S. Ren 和 J. Sun，“用于图像识别的深度残差学习”，载于IEEE计算机视觉与模式识别会议论文集，2016年，第770–778页。[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser 和 I. Polosukhin，“Attention Is All You Need”，载于神经信息处理系统会议论文集，2017年，第5998–6008页。[26] X. Han, Y.-T. Wang, J.-L. Feng, C. Deng, Z.-H. Chen, Y.-A. Huang, H. Su, L. Hu 和 P.-W. Hu，“基于Transformer的多模态预训练模型综述”，《神经计算》，第515卷，第89–106页，2023年。 [在线]。获取地址：https://www.sciencedirect.com/science/ article/pii/S0925231222012348 [27] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo 和 S. Gould，“VLN BERT：用于导航的循环视觉与语言BERT”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2021年，第1643–1653页。[28] X. Li, C. Li, Q. Xia, Y. Bisk, A. Celikyilmaz, J. Gao, N. A. Smith 和 Y. Choi，“结合语言预训练和随机采样的鲁棒导航”，载于2019年自然语言处理实证方法会议与第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集，2019年，第1494–1499页。[29] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva 和 D. Batra，“DD-PPO：利用25亿帧学习近乎完美的PointGoal 导航器”，ICLR论文集，2019年。[30] F. Zhu, Y. Zhu, X. Chang 和 X. Liang，“结合自监督辅助推理任务的视觉-语言导航”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2020年，第10 012–10 022页。[31] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik 等，“Habitat：具身人工智能研究平台”，载于IEEE/CVF ICCV论文集，2019年，第9339–9347页。[32] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva 等，“具身导航智能体的评估”，arXiv预印本 arXiv:1807.06757，2018年。[33] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh 和 D. Batra，“利用点云感知在照片级真实感环境中进行具身问答”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2019年，第6659–6668页。[34] S. Ross, G. Gordon 和 D. Bagnell，“将模仿学习和结构化预测化简为无悔在线学习”，载于AISTATS论文集。JMLR研讨会与会议论文集，2011年，第627–635页。

[35] J. Schulman、F.Wolski、P. Dhariwal、A. Radford 和 O.Klimov，“近端策略优化算法”，*arXiv* 预印本 *arXiv:1707 06347*，2017年。。