

IMUNet: Efficient Regression Architecture for IMU Navigation and Positioning

摘要

近年来，基于数据驱动的方法在导航和定位领域备受关注，并在准确性和效率方面优于所有竞争方法。本文介绍了一种新的架构——IMUNet，该架构能够在接收原始 IMU 测量数据序列的情况下，实现高精度且高效的边缘设备位置估计。该架构与近期为边缘设备实现而引入的最新一维卷积神经网络 (CNN) 版本进行了对比，在准确性和效率方面表现出色。此外，本文提出了一种利用手机中的 IMU 传感器和 Google ARCore API 的新数据集采集方法，并记录了一个公开可用的数据集。通过使用四个不同的数据集以及所提出的数据集和实际设备的实现，进行了全面的评估，以验证该架构的性能。为了促进后续研究，所有代码，包括 PyTorch 和 TensorFlow 框架下的实现代码，以及 Android 应用程序代码，均已公开共享。

关键词：深度学习；数据驱动方法；惯性导航；IMU 测量

1 引言

三轴陀螺仪和三轴加速度计构成了常见的惯性测量单元 (IMU)。这些传感器可以测量移动平台的旋转速度和线性加速度。某些 IMU 传感器还能够提供移动平台的重力补偿线性加速度。由于 IMU 能够无需外部基础设施采集运动数据，因此被广泛应用于各种导航系统和应用中，包括边缘设备应用 [1]、机器人 [2]、无人机 [3]、自动驾驶车辆 [4] 等。

目前，大多数 IMU 数据处理方法依赖于手工设计的模型，以近似传感器特性和底层运动的动态特性。在文献 [5] 中，总结了 IMU 数据处理的流程，包括：数据过滤、补偿传感器固有特性的模型、通过插值和外推技术进行数据估计、用于姿态计算的积分技术，以及用于减轻测量漂移的传感器融合技术。数据过滤和模型补偿被称为预处理步骤，主要用于通过消除传感器固有特性引起的误差或测量过程中收集的噪声，从而获得干净的数据。

插值、外推和积分技术是基于底层运动动力学确定 IMU 集成姿态和位置的常见方法。然而，由于 IMU 积分过程中不可避免的漂移问题，额外的传感器数据 [6] [3] 或运动约束 [7] [8] 通常被结合使用，通过概率方法更新传感器数据以减轻测量漂移。

当前 IMU 数据预处理方法适用于生成无噪声的数据供后续处理。然而，它们仍然容易受到某些误差来源的影响，其中最重要的之一是传感器建模误差。在文献 [7] [9] 中，作者研究了 IMU 传感器的信号处理模型，而文献 [10] [11] 则探讨了传感器建模误差的固有模型。然而，这些模型是手工设计的，受到工程限制的影响，无法作为所有平台和应用的通用解决方案。

本文提出了一种基于数据驱动策略的新型机器学习架构，用于惯性传感器建模，以实现更高的性能和效率。这一灵感来源于最近采用数据驱动方法和神经网络进行定位、导航和时间测量的研究 [8] [12]- [13]。本文设计了一种适用于边缘设备的深度神经网络，该网络可以处理和融合 IMU 数据测量，并将位置和速度作为输出。此外，本文提出了一种基于 Android 手机收集数据集的方法，并采集了一个全新数据集。该数据集采集方法是对文献 [14] 中方法的改进，与文献 [15] 中的方法类似，但区别在于本文未使用 Tango 设备，而是通过 Google ARCore API（由 Google 于 2019 年推出 [16]）获取地面真实数据。通过这种方式，用户可以使用当前的 Android 手机采集数据集。

除了提出的架构，本文还评估了一些边缘设备友好型 CNN 架构（如 MobileNet [17]、MobileNetV2 [18]、MnasNet [19] 和 EfficientB0 [20]），并将这些架构修改为与 IMU 测量数据兼容。随后，这些模型被用作最近成功的数据驱动方法 [21] 的回归模型。本文在多个数据集上评估并报告了这些模型、提出的模型，以及 ResNet18 架构 [22]（文献 [21] 中使用的模型）的性能表现。

2 相关工作

IMU 传感器容易受到多种误差来源的影响。这些误差通常由电气、机械问题以及信号处理系统的缺陷引起。为了降低 IMU 数据中的噪声并调整固有模型中的误差，统计方法是目前最常用的策略。在惯性导航方法中，通常通过滤波来消除或衰减 IMU 测量输出中的高频成分 [9]。例如，文献 [23] 中作者使用了自回归与移动平均（ARMA）方法来实现这一目的。此外，为了应对除噪声之外影响 IMU 测量的偏差，还可以在测量方程中引入额外的校正分量 [10]。

尽管在高端 IMU 传感器中，仅处理噪声和传感器偏差通常可以满足精度要求，但对于低成本 IMU 传感器，需要考虑更多因素，如 G 灵敏度矩阵、未对齐误差和比例因子 [24] [25]。文献 [26] 中，作者还考虑了热误差对实际应用准确性的影响。根据牛顿运动定律 [6] [5] 和数值积分，当获取到清晰且无误差的测量数据时，可以计算出惯性位置。然而，由于误差高度依赖于 IMU 传感器类型、信号处理方法和测量程序，单纯的数学建模方法难以获得完全干净的数据。

即使对噪声有适当的数学描述，在姿态积分中仍然会出现漂移。为了解决累积漂移问题，一种方法是将 IMU 数据与其他传感器的数据融合，例如视觉传感器 [6]、全球定位系统 (GPS) [5]、轮式里程计 [2] 和激光雷达 [27]。另一种方法是通过施加运动约束或考虑平台的重复运动模式，使用概率技术来减小估计误差 [7] [8] [28]。例如，“零速度事件”方法 [7] [8] 将 IMU 安装在足部，并利用脚接触地面时速度为零的约束来纠正漂移。然而，这些方法需要额外的足部传感器，对用于边缘设备的 IMU 传感器以及无人机导航场景不适用。

2.1 步态模式方法

步态模式方法 [28] 通过考虑平台的重复运动模式来估计导航。例如，在人类运动等本质上具有重复性的运动中，可以利用运动的启发性规律来实现导航估计。步数计数法是一种基于条件来估算追踪和导航的方法，其假设追踪距离与步数相关，IMU 传感器在追踪期间提供稳定数据。在这种受控环境下，结果通常具有较好的准确性和令人印象深刻的表现。文献 [29]

中，作者尝试通过频域分析估计运动方向，而其他研究者则结合了受控运动和附加传感器数据。例如，文献 [30] 中，Kottas 等人通过结合悬停运动检测模块提出了一种视觉惯性里程计 (VIMO) 方法，它利用多个传感器的输入来估计传感器姿态。尽管这些方法表现出色，但它们并未体现出基于数据驱动方法的鲁棒性。

2.2 数据驱动方法

近年来，数据驱动策略作为一种解决积分过程中累积漂移的新方法，受到了广泛关注。深度神经网络 (DNNs) 在多个应用中表现优异，这启发了研究者利用这些方法构建 IMU 模型来提高惯性导航精度 [1] [8]。这些模型通过对 IMU 传感器数据进行端到端的处理，估计其姿态 [21] [13] [15]。此外，这些模型还被用来表征传感器模型 [31] [32]。例如，文献 [33] 中采用循环神经网络 (RNN) 模型作为回归模型来估算位置，这种模型尤其适合处理时间序列数据，并能将记忆信息从前一阶段传递到后续点。IMU、磁力计和气压计的测量值被用作输入，输出为位置和速度。

所有数据驱动方法（包括文献 [21] 中提出的方法）的性能高度依赖于回归模型，以及其在误差修正、噪声消除、传感器校准甚至导航方程方面的能力，这些能力体现在模型的架构和参数中。更重要的是，对于端到端的解决方案，这些方法需要在实时环境下表现良好，因此模型的推理时间、大小和能耗是关键因素。例如，在使用手机设备和仅依赖 IMU（惯性测量单元）传感器进行惯性导航时，较小的模型尺寸和较短的推理时间在移动实现中更有优势，也更适用于实时应用。IMU 传感器可以在手机上运行超过 24 小时而不会引起电池耗尽问题，而更适合的模型能够进一步减少能耗。

本文研究了多种为边缘设备优化的最新 CNN 模型，包括：

- *Mobilenet* [17]
- *MobilenetV2* [18]
- *MnasNet* [19]
- *EfficientNetB0* [20]

1) *Mobilenet* [17]: 2017 年提出的 Mobilenet 是一种为移动和嵌入式实现设计的卷积神经网络 (CNN) 架构。该架构采用了深度卷积 (depth-wise convolution) 和点卷积 (point-wise convolution) 代替传统的卷积操作，这大大降低了计算成本，使得该架构更适合嵌入式设备的实现。深度卷积和点卷积是传统卷积的因式分解卷积。在深度卷积中，单个卷积核分别应用于输入的每个通道；在点向卷积中，使用 1×1 卷积来合并深度卷积的所有输出。而传统卷积操作同时进行过滤和合并结果到新的输出集合中。由于这种新的架构将过滤操作与结果合并分开，计算成本和模型大小被大大减少。

2) *MobilenetV2* [18]: 在 [18] 中，作者对 Mobilenet 架构进行了小幅改进。其理念是结合 Mobilenet 中的点卷积和深度卷积，并在新架构中使用残差块 (residual block)。在传统的残差块中，残差连接将较大的通道连接在一起。然而，在 MobilenetV2 中，残差连接是从一个窄的或瓶颈通道连接到下一个窄通道，这种结构被称为反向残差块 (inverted residual block)。在卷积操作中，MobilenetV2 采用了 Mobilenet 的思想，通过使用深度卷积和点卷积来减少

模型的大小和计算成本。此外，作者证明了非线性会消除每个反向残差块到下一个块的信息，因为窄卷积核已将其连接在一起。因此，在每个反向残差块的末端没有使用非线性函数，整个块被称为反向残差块和线性瓶颈（linear bottleneck）。

3) *MnasNet* [19]: 手动为特定目的设计 CNN 模型是非常困难的，因为在设计时需要平衡相互排斥的目标，而可设计的神经架构种类繁多，调整这些平衡是一项繁琐的工作。为了解决这个问题，MnasNet 与 Mobilenet 和 MobilenetV2 不同，它是通过一种名为神经架构搜索（Neural Architecture Search, NAS）的方法自动发现的。作者提出了一种方法，通过在实际设备上执行推理，平衡模型的准确性与推理延迟之间的权衡。在搜索空间方面，引入了一种新的因式分解层次搜索空间。这个搜索空间允许网络中不同层之间的多样性，从而提高了架构设计的灵活性。

4) *EfficientNetB0* [20]: 这个模型也是通过神经架构搜索技术发现的。与 [19] 中的方法相同，使用了相同的搜索技术。然而，该方法并不专注于硬件设备，作者的主要目标是找到一个基础网络并将其扩展，以便发现其他架构。因此，模型的准确性与 FLOPS（每秒浮点运算次数）数量之间进行了权衡。由于该模型是该方法中最小的模型，因此我们仅将该模型从 EfficientNet 系列中作为比较对象。

3 本文方法

本文的实验表明，所有提到的模型和 [21] 中使用的 ResNet18 模型在不同数据集上的表现接近。然而，这些模型在移动端实现时的大小和推理时间有所不同。ResNet18 使用常规卷积，而其他所有模型都使用深度卷积和逐点卷积来替代常规卷积。受 Mobilenet [17] 启发，本文通过实验发现，将 ResNet18 中的卷积层替换为深度卷积和逐点卷积，不仅能大幅降低模型的计算成本，还能提高模型的准确性。由于深度卷积是在每个通道上分别进行的（这里是不同维度的 IMU 测量数据），每个 IMU 测量的噪声不会在卷积过程中影响其他 IMU 的输出。通过防止卷积过程中的干扰，更多纯净的信息可以传递到后续层，从而提高准确性。此外，残差块的设计能够将输入传递到输出块，转换纯净的数据并防止过拟合。本文提出了一种新的块——MobileResNet 块。在这个块中，使用了深度卷积、逐点卷积和批量归一化，旨在减少计算成本并减少不同通道之间的噪声干扰。同时使用残差连接来避免过拟合。其余架构类似于 [21] 中介绍的 ResNet18，它是 [22] 的一维版本。此外，本文使用了 [34] 中介绍的指数线性单元（ELU）作为激活函数，替代了 ReLU。ELU 能够防止“死神经元”问题，减少训练时间并提高准确性。然而，根据公式 1，对于负值，ELU 的指数计算速度比 ReLU 及其导数要慢，从而增加了推理延迟。然而，由于提高了准确性，这一延迟可以忽略不计。图 1 展示了 MobileResNet 块的架构。IMUNet 架构的详细信息已在表 1 中给出。

$$y = \begin{cases} x & \text{if } x \geq 0 \\ \exp(x) - 1 & \text{if } x < 0 \end{cases} \quad (1)$$

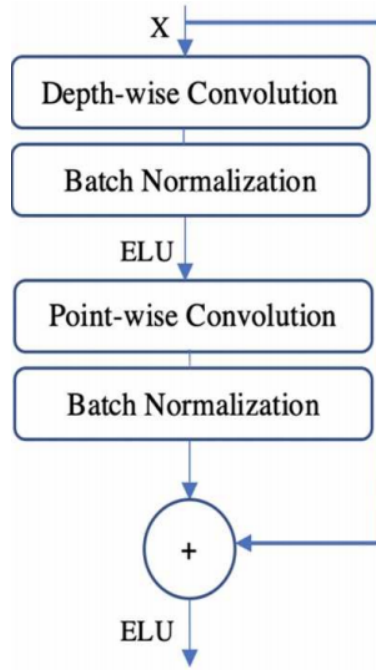


图 1. MobileResnet 块示意图

表 1. IMUNet 的架构. 其中 m 是维度数

<i>Stage</i>	<i>Operator</i>	<i>Resolution</i>	<i>Channels(n)</i>	<i>Layer</i>
1	Input	6*200	6	-
2	Conv1D	2*64	64	1
3	MRBlock	2*64	64	2
4	MRBlock	1*128	128	2
5	MRBlock	1*256	256	2
6	MRBlock	1*512	512	2
7	Conv1D	1*128	128	1
8	Dense	1*m	512	2

4 复现细节

4.1 与已有开源代码对比

- **尝试减少 IMUNet 的层数并测试：**
在论文设计的 IMUNet 中**减少**了一个 MRBlock 层，并在某个数据集上进行测试。
- **尝试增加 IMUNet 的层数并测试：**
在论文设计的 IMUNet 中**增加**了一个 MRBlock 层，并在某个数据集上进行测试。

4.2 实验环境搭建

操作系统使用的是 Windows10 专业版。Python 版本使用的是 3.11.9, cuda 版本为 12.4, pytorch 版本为 2.4.1. 同时 python 中安装了 tensorflow-lite 以及 onnx 库, 用于将 pytorch 模型转换为 tf-lite 模型, 以便进行边缘设备的移植。使用的显卡型号为 NVIDIA Geforce RTX 3090.

5 实验结果分析

为了评估所提出的模型, 本文进行了多个实验, 在四个不同的公开数据集以及文中所提出的数据集上进行了评估。本文将这五个数据集应用于数据驱动方法, 使用了提到的最先进的机器学习模型以及文中所提出的架构。对于 RONIN 数据集, 使用了其中一半公开的数据。OXIOD 和 RIDI 数据集中的所有数据均已被使用。为了评估模型在三维数据以及复杂机动中的表现, 本文还下载了一些 PX4 数据集的日志, 并将其应用于该框架。还使用了所提出的数据集, 这个数据集结合了使用 ARCore 和 Tango 设备收集的地面真值数据, 用于评估架构。

实验评估使用了 Python 语言。所有实验中都使用了 [21] 中使用的绝对轨迹误差 (ATE) 和相对轨迹误差 (RTE)。ATE 是均方根误差 (RMSE), RTE 是预定时间间隔内 ATE 的平均值。所有模型的训练都使用了单个 GPU (Nvidia GeForce GTX 3090)。使用了与原文相同的参数, 但每个模型都训练了 300 个 epoch。

为了评估所有模型的效率, 我们考虑了两个常见的度量标准: 参数数量和 FLOPs (浮点运算次数)。尽管参数数量能够提供对模型效率的一般解释 (模型尺寸越小, 意味着内存引用越少, 能效越高), 但它并不能准确反映延迟推理过程中乘法运算的次数。因此, 我们同时考虑了 FLOPs 和参数数量来评估每个模型的效率。图 2 和图 3 分别展示了所有模型的 FLOPs 和参数数量。结果表明, 所提出的模型在准确性和效率方面均优于所有现有的最先进模型。

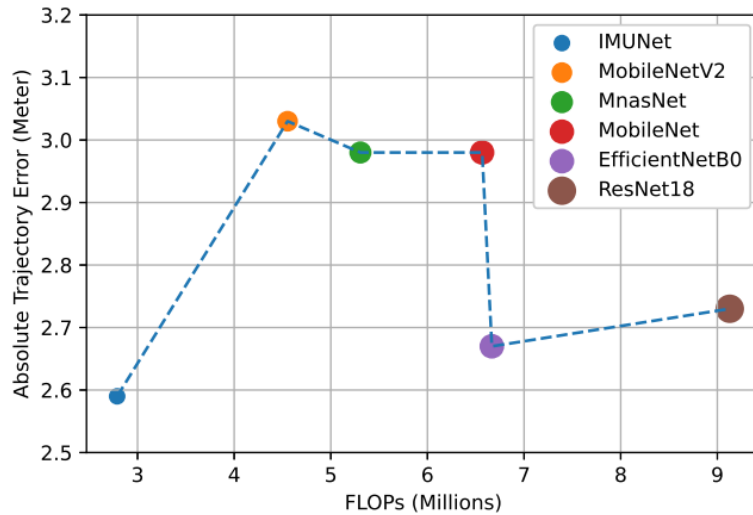


图 2. 每种架构所提出的数据集的浮点运算数 (FLOPs) 数量和准确度

图 4 展示了使用论文原有的 IMUNet 对某个数据集中的某个对象进行测试的结果, 图 5 和图 6 分别展示了在 IMUNet 中**增加**一个 MRBlock 层以及在 IMUNet 中**减少**一个 MRBlock 层的结果, 可以发现原有的 IMUNet 的误差最小, 其确实是合适的网络。

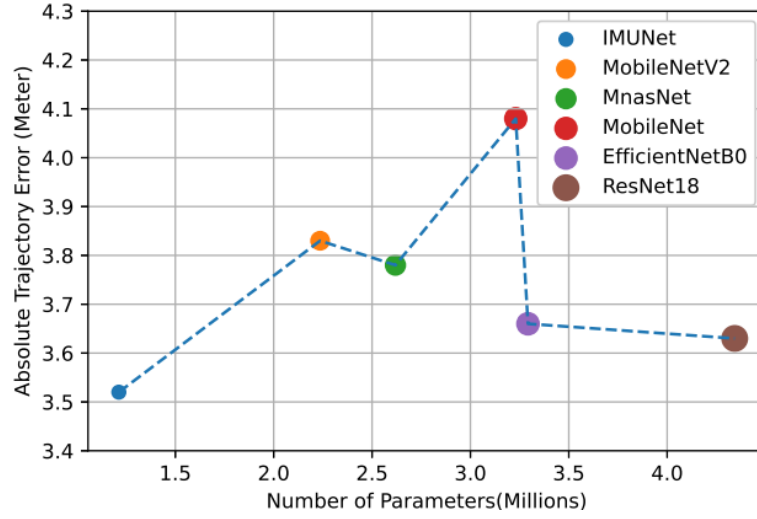


图 3. RONIN 数据集上的网络参数数量和精度

每个数据集中，选择了测试集中的示例轨迹，并在图 7 和图 8 中展示了所有网络的性能。未见过的轨迹指的是在训练阶段未包含的样本。对于 PX4 数据集，受试者来自不同的人，因此这些被认为是未见过的受试者。从图表和表格中可以看出，所提出的模型在准确性方面优于现有的边缘设备友好型架构。

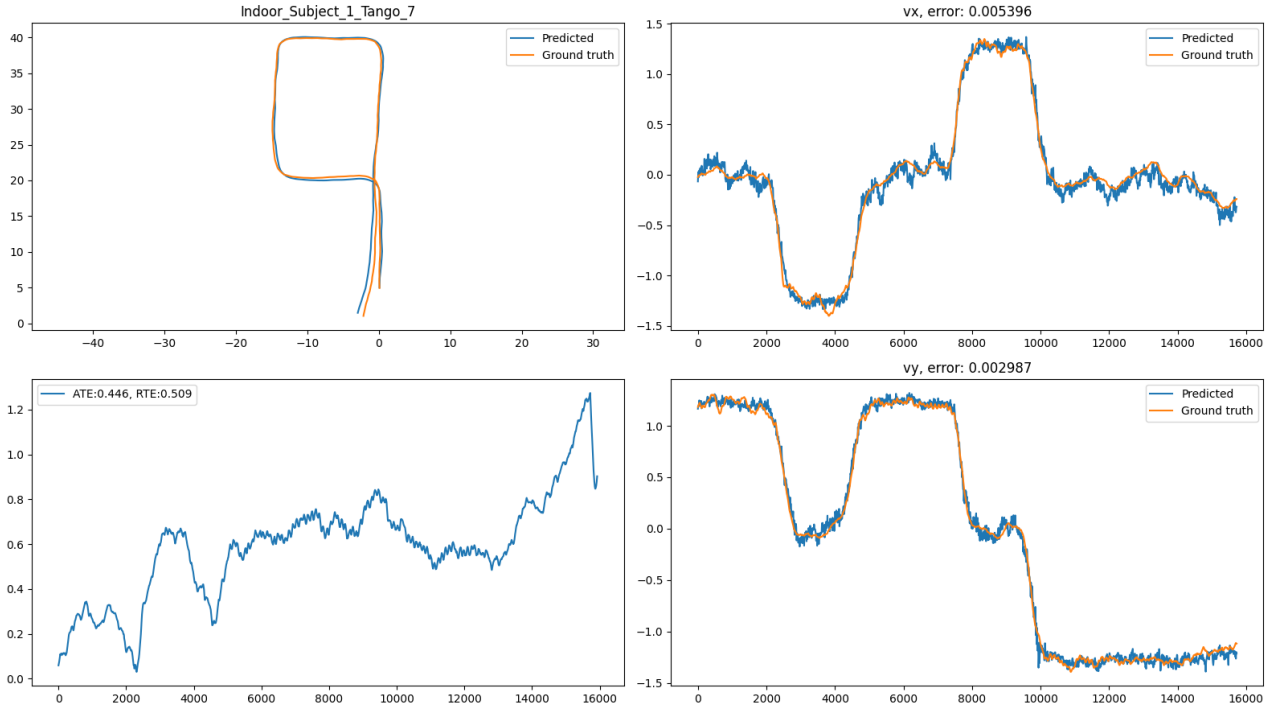


图 4. 未改动的 IMUNet

本文在边缘设备上使用提出的数据集实现了 RONIN 方法的测试部分，以评估所有模型的实时实现性能。为此，所有模型的 TensorFlow Lite 版本已被创建。对于 Pytorch 模型，我们使用了 ONNX 库，这是一个第三方库，用于将深度学习模型从 Pytorch 转换为 Keras 版本。然而，实验结果中我们使用了 Keras 实现的模型。实验使用的设备为 Galaxy S10 手机。该应用程序还包含使用 ARCore API 收集数据集的方法。表 2 展示了每个模型及提出模型的

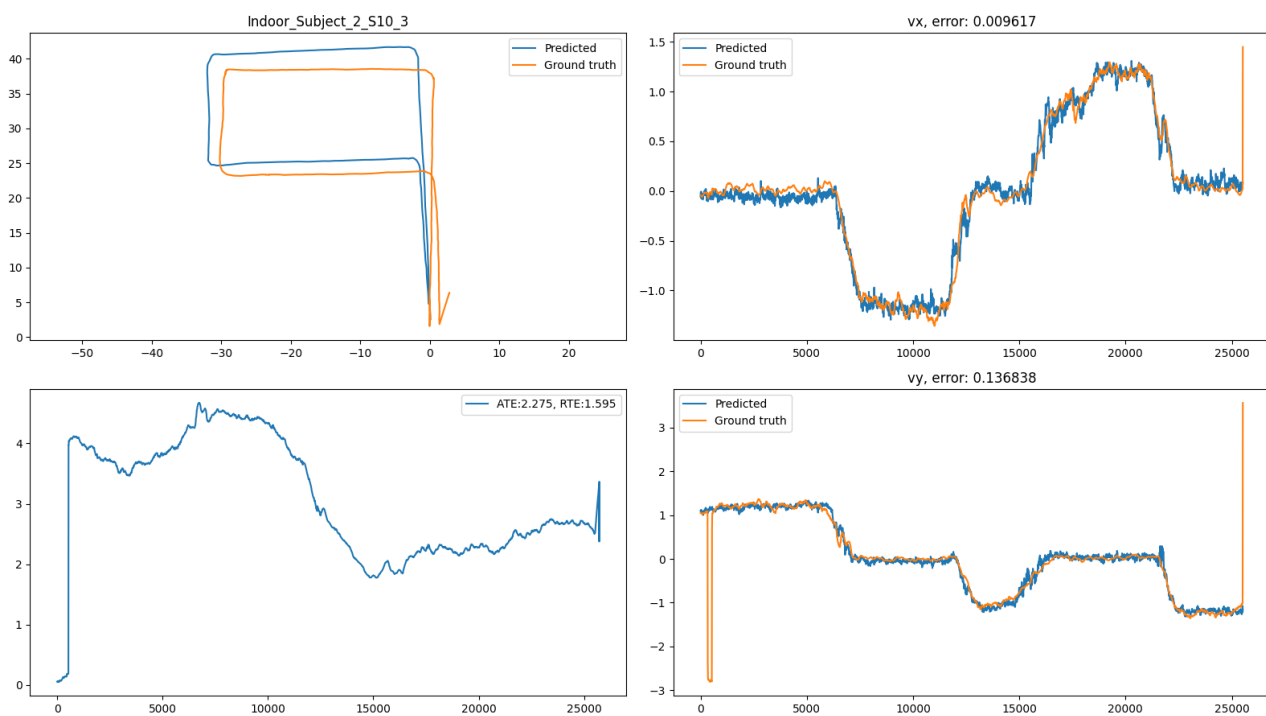


图 5. 增加一个 MRBlock 层的 IMUNet

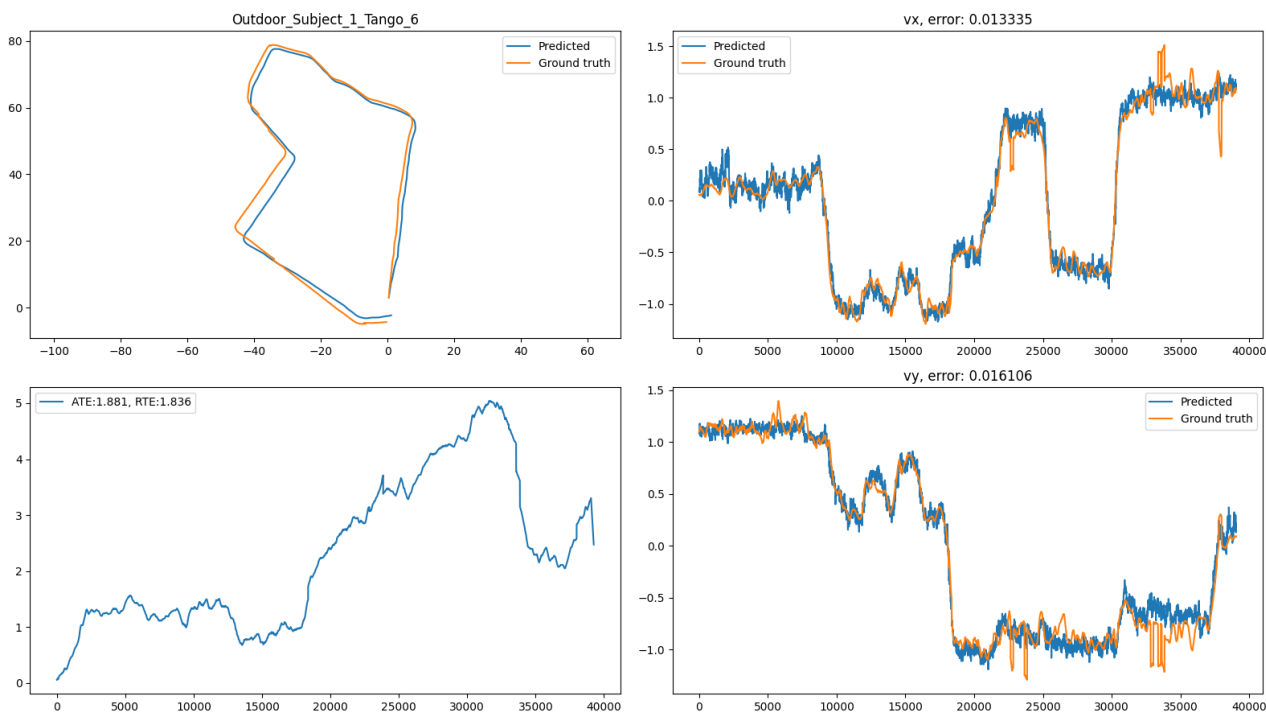


图 6. 减少一个 MRBlock 层的 IMUNet

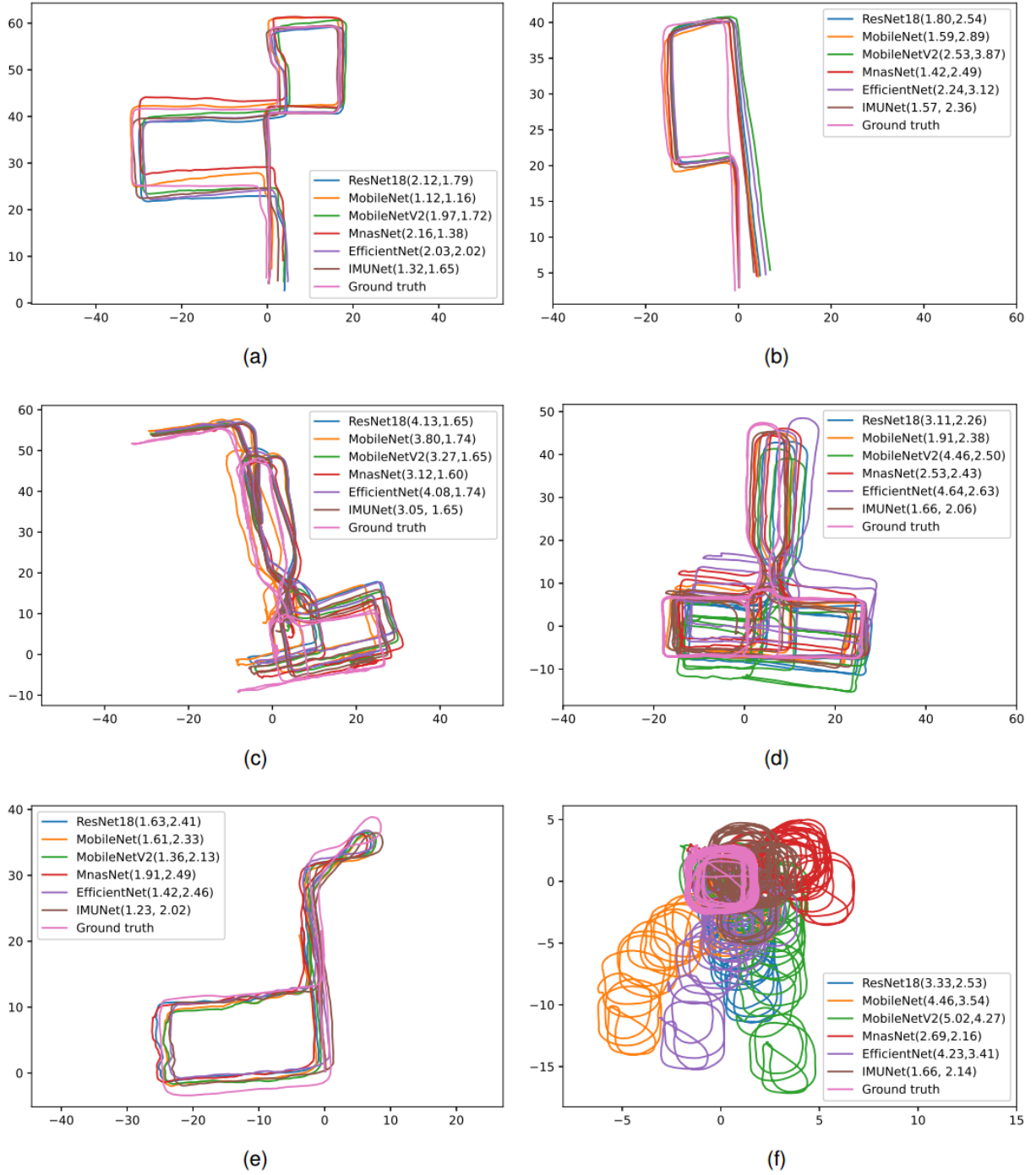


图 7. 来自文中所提出的、RONIN、RIDI 和 OXIOD 数据集的轨迹，以及所有最先进网络的性能。括号内的数字分别以米为单位表示绝对弹道误差 (ATE) 和相对弹道误差 (RTE)。所有轴的单位都是米。(a) 使用 Tango 设备的拟议数据集的轨迹。(b) 使用 ARCore API 进行地面真值收集的拟议数据集的轨迹。(c) 来自 Seen 数据的 RONIN 数据集的轨迹 (环境与训练集相同)。(d) 来自 Unseen 数据的 RONIN 数据集的轨迹 (环境与训练集不同)。(e) RIDI 数据集的轨迹。(f) 来自 OXIOD 数据集的轨迹。

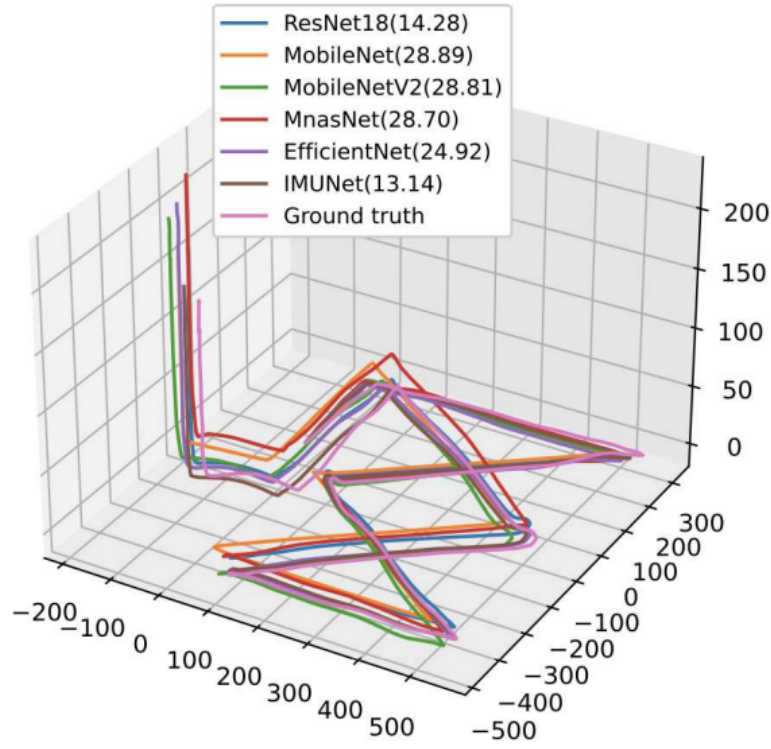


图 8. 来自 PX4 数据集的轨迹图以及所有当前最先进网络的性能（括号中的数字表示绝对轨迹误差 (ATE)，单位为米。轴的单位也是米）

延迟推理时间和 TensorFlow Lite 模型大小。提出模型的推理时间和模型大小均小于其他网络，显示出其在节能和实时实现方面的优势。

表 2. 所有 SOTA 模型在实际边缘设备上的延迟推断和 tensorflow-lite 模型大小

	Metric	ResNet18	MNet	MNetV2	MnasNet	EffNet	IMUNet
tfLite Size	Mb	4.5	3.5	2.7	3.1	3.8	1.4
Latency	μSec	1044	907	645	654	967	387

6 总结与展望

本文介绍了一种新的惯性导航架构，利用 IMU 测量序列进行导航。神经惯性导航方法是执行导航和定位的最可靠方法。这些方法的性能高度依赖于神经网络的能力。更重要的是，由于最终目标是使导航能够在边缘设备上实现，因此架构的效率对于实时实现至关重要。本文提出了一种精确且高效的网络，并介绍了一种新的数据集收集方法，并分享了代码，使任何人都可以使用手机收集数据集以进行进一步研究。通过实现和应用一种适用于边缘设备的、面向惯性导航目的的一维版本的最先进卷积神经网络，进行了经验性研究。所有代码均已公开分享，供修改和进一步研究使用。

参考文献

- [1] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, “Tlio: Tight learned inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.
- [2] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, “Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization,” *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1081–1099, 2021.
- [3] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, “Vimo: Simultaneous visual inertial model-based odometry and force estimation,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.
- [4] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 163–168.
- [5] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.
- [6] M. Li and A. I. Mourikis, “High-precision, consistent ekf-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [7] K. Abdulrahim, T. Moore, C. Hide, and C. Hill, “Understanding the performance of zero velocity updates in mems-based pedestrian navigation,” *International Journal of Advancements in Technology*, vol. 5, no. 2, 2014.
- [8] B. Wagstaff and J. Kelly, “Lstm-based zero-velocity detection for robust inertial navigation,” in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2018, pp. 1–8.
- [9] C.-H. Kang, S.-Y. Kim, and C.-G. Park, “Improvement of a low cost mems inertial-gps integrated system using wavelet denoising techniques,” *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 4, pp. 371–378, 2011.
- [10] N. Trawny and S. I. Roumeliotis, “Indirect kalman filter for 3d attitude estimation,” *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.
- [11] T. Schneider, M. Li, M. Burri, J. Nieto, R. Siegwart, and I. Gilitschenski, “Visual-inertial self-calibration on informative motion segments,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6487–6494.
- [12] C. Chen, X. Lu, A. Markham, and N. Trigoni, “Ionet: Learning to cure the curse of drift in inertial odometry,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

- [13] M. Zhang, M. Zhang, Y. Chen, and M. Li, “Imu data processing for inertial aided navigation: A recurrent neural network based approach,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3992–3998.
- [14] S. Cortés, A. Solin, E. Rahtu, and J. Kannala, “Advio: An authentic dataset for visual-inertial odometry,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 419–434.
- [15] H. Yan, Q. Shan, and Y. Furukawa, “Ridi: Robust imu double integration,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 621–636.
- [16] “Google. arcore.” [Online]. Available: <https://developers.google.com/ar/>
- [17] A. G. Howard, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [19] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, “Mnasnet: Platform-aware neural architecture search for mobile,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.
- [20] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.
- [21] S. Herath, H. Yan, and Y. Furukawa, “Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods,” in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3146–3152.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [23] A. G. Quinchia, G. Falco, E. Falletti, F. Dosis, and C. Ferrer, “A comparison between different error modeling of mems applied to gps/ins integrated systems,” *Sensors*, vol. 13, no. 8, pp. 9549–9588, 2013.
- [24] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, “High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 409–416.
- [25] Y. Yang, P. Geneva, X. Zuo, and G. Huang, “Online imu intrinsic calibration: Is it necessary?” *2020 Robotics: Science and Systems*, 2020.

- [26] X. Niu, Y. Li, H. Zhang, Q. Wang, and Y. Ban, “Fast thermal calibration of low-grade inertial sensors and inertial measurement units,” *Sensors*, vol. 13, no. 9, pp. 12 192–12 217, 2013.
- [27] P. Geneva, K. Eickenhoff, Y. Yang, and G. Huang, “Lips: Lidar-inertial 3d plane slam,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 123–130.
- [28] A. Ahmed and S. Roumeliotis, “A visual-inertial approach to human gait estimation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4614–4621.
- [29] M. Kouroggi and T. Kurata, “A method of pedestrian dead reckoning for smartphones using frequency domain analysis on patterns of acceleration and angular velocity,” in *2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014*. IEEE, 2014, pp. 164–168.
- [30] D. G. Kottas, K. J. Wu, and S. I. Roumeliotis, “Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3172–3179.
- [31] C. Jiang, S. Chen, Y. Chen, B. Zhang, Z. Feng, H. Zhou, and Y. Bo, “A mems imu de-noising method using long short term memory recurrent neural networks (lstm-rnn),” *Sensors*, vol. 18, no. 10, p. 3470, 2018.
- [32] M. Brossard, S. Bonnabel, and A. Barrau, “Denoising imu gyroscopes with deep learning for open-loop attitude estimation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4796–4803, 2020.
- [33] A. AbdulMajuid, O. Mohamady, M. Draz, and G. El-bayoumi, “Gps-denied navigation using low-cost inertial sensors and recurrent neural networks,” *arXiv preprint arXiv:2109.04861*, 2021.
- [34] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus). arxiv 2015,” *arXiv preprint arXiv:1511.07289*, 2020.