4 Discussion

一个上手简单，逻辑清晰的模型开发框架能够使更多的专业人士结合自己的学识开发出独特且优越的模型，就好比torch，tensorflow等深度学习框架，正是这些框架的支撑，深度学习的研究得以蓬勃发展。水文模型的开发同样离不开一个高效、灵活、易上手的开发框架。本节将针对LumpedHydro.jl框架的特征进行探讨，对模型灵活性、高效性等特性进行评估。

4.1 framework feature

（1）easy using

LumpedHydro.jl框架旨在为水文模型应用和开发的人员都提供便利可靠的支持。对于想要应用水文模型进行预报工作的需求，使用者可以直接调用框架中所提供的成熟的水文模型，同时模型的输入可以从txt, csv等通用文件中读取，并构建NamedTuple类型作为模型输入，无需额外的输入文件整合；针对水文模型的开发需求，使用者可以选择根据水文模型的计算公式和结构组成，从Flux到Element再到Unit依次构建水文通量，水文模块和水文模型，有着以名称驱动下严谨、具有逻辑性的构建思路；针对水文模型的修改需求，使用者可以根据update, add, remove方法更改Flux计算公式或element的输入输出水量平衡。最后在完成模型模拟计算后，使用者能够得到NamedTuple类型的计算结果，这里面包含了径流模拟结果在内的水文模型中所有的中间计算通量，使用者无需额外调用函数来获取。

（2）flexible

模型框架的灵活性是水文模型灵活运用与开发的基础，得以Flux, element和Unit等结构的设计，使用者能够根据模型计算公式快速、自由的构建一个新的水文模型。框架灵活的基础则是得益于以namedtuple为中间计算结果的水文通量的存储与调用，因此在满足输入输出通量匹配的前提下，模型就可以实现灵活拼接，构建出多种类型的水文模型。然而灵活性与模型的计算效率是存在一个trade-off，例如，使用框架搭建ODE问题需要重复调用并获取flux的计算结果，这之中的计算结果存储与读取会使得问题的求解效率大打折扣。为此该框架针对这个问题，在模型构建的过程中就采用匿名构建方式实现ODE计算函数的生成，从而避免中间计算结果的存储与调用，在保证灵活性的前提下仍有着优秀的计算性能。

(3) reusable

水文模型发展以来诞生了大量成熟应用广泛的模型，包含了融雪、蒸发、产流、下渗等水循环过程的计算公式以及融雪模块，土壤计算模块和壤中流模块等计算模块，当构建一个新的水文模型时，我们可以参考并使用一些经典模型的计算公式和模块，随机组合各模型的计算公式与模块，为水文预报领域带来无限可能。为此LumpedHydro.jl框架参考了MARRMoT按不同通量类型构建计算公式，使用者可以根据输入输出和参数名称实现这些公式的调用（基于julia的多重派发特性）。

(4) decoupled

LumpedHydro.jl框架的解耦特征在这里指的是模型构建与参数设定的解耦，通常而言，在一些框架中构建模型的同时需要给出模型参数的实际值，而LumpedHydro.jl则将两者进行拆分，即构建模型无需设定模型参数值，而是与数据输入一同作为模型模拟的输入argument，这种设计思想不仅能够简化模型构建过程，支撑模型的复用性，还避免在参数率定过程中重复构建模型，降低率定的计算效率。

(5) differential

（6）support more modern models and algorithms(SciML Ecosystem **scalability**)

LumpedHydro.jl框架中PINN模型构建、ODE问题求解与参数率定两个核心功能均是依托于SciML的Lux.jl，DifferentialEquation.jl和Optimization.jl等包，因此可以很好的利用SciML提供的成熟生态和高效的求解器完成PINN模型开发、径流模拟和参数率定。

4.2 framework drawback and future development

（1）仅针对于集中式模型构建，但是未来会开始考虑基于lumpedhydro.jl的spatial计算

Lumped 水文模型由于无法捕捉大流域的空间异质性、洪水演进等特性，通常仅能够应用于小流域的洪水预报中，其应用范围受限。为此在后续工作中，需要以Lumpedhydro.jl为支撑，以该包构建各个计算单元或子流域的水文模型，再通过河流拓扑关系或者水流流向关系构建半分布式和分布式水文模型，实现dPL-HBV的节点中多模型优化，river-network, dPL-distributed等模型。大型流域的分布式水文模型计算中往往涉及并行化、GPU加速等技术提升模型的模拟效率，而在lumped 水文模型开发时我们并没有考虑GPU来加速计算效率，因为基于julia语言卓越的计算性能，一个简单的水文模型能够在毫秒尺度完成模拟。然而，分布式水文模型中大规模计算单元的计算需求使得LumpedHydro.jl需要进一步考量如何充分利用julia语言的计算性能，包括StructArray.jl，GPU加速，元编程等。

（2）更灵活的element重利用

水文预报研究发展至今已经开发出大量水文模型，伴随而来的是大量描述水量平衡的计算模块，例如cemaneige，exphydro和hbv等模型均存在不同的融雪计算模块。尽管在实现各个水文模型时，我们会尽可能地将其划分成多种计算模块，如融雪层计算，土壤层计算，壤中流层，线性水库层。然而， element的遍历计算依赖于变量名称，若是拼接来源于不同模型的计算模块，其变量名称很难匹配，为此在拼接计算模块前需要额外开展变量转换的工作。因此在未来工作中可以针对水文模型结构之间的共性和典型水文通量和模块，开发出一种统一的模型模板，约束各种类型模块的输入输出通量，以此支撑不同模型的计算模块的拼接。

（3）图形化编程

为了向水文预报工作者或者水文模型研究者提供更便捷的使用方式，一些水文模型软件通常有着用户优化的图形化界面，使用者可以仅通过点击并上传文件实现预报工作。而为便利使用者能够仅通过输入公式构建flux，并且通过拓扑连接实现element以及unit的构建，按特殊格式保存并应用于模拟和率定工作。