具体使用方法和预训练模型及其训练数据，都可以在DAP-2预训练模型官网找到：

<https://aissquare.com/models/detail?pageType=models&name=DPA-2.2.0-v3.0.0b3&id=272>

目前脚本和预训练模型都是基于DeePMD-Kit-v3.0.0b3，请务必确认预训练模型版本和DeePMD-Kit版本一致。

<https://github.com/deepmodeling/deepmd-kit/releases/tag/v3.0.0b3>

简单使用方法：

①描述符的参数不要动。如果更换预训练模型，请用parma.py脚本输出预训练模型的参数，再根据输出参数自行更改。我提供的json脚本已经是符合提供的预训练模型的参数，可直接使用。

②Mutitak的json和普通任务的json核心上的一致的，只是model\_dict和loss\_dict变成了多个分支的组合。从model\_dict模块开始可以按需修改我们的任务。实例如图：



“pureknachar”是我的任务，这部分可以根据你的实际情况进行修改。“Domian\_ANI”是预训练模型中的branch，在我提供的DAP-2预训练模型官网中，具体给出了不同branch的名称和训练细节。可以根据自己实际任务进行选择。后面还会有loss\_dict，根据model\_dict一一对应进行修改即可。

③训练数据准备

预训练模型的训练数据也可以在官网得到。里面都是已经转为dpdata格式的文件。如果在Multitask中想要用这些数据辅助训练（我也非常推荐用它们进行辅助训练，避免在训练自己分支的时候过拟合），可以使用我提供的path.py脚本，读取并输出当前目录下所有dpdata的路径并且符合json输入格式要求到path.txt中。split.py脚本可以将path.txt分割训练集和验证集。直接复制即可使用。

④训练或者微调（finetune）

如果要直接训练，直接使用：dp --pt train xxxx.json即可。如果是要微调，则在后面加上finetune命令即可：dp --pt train xxxx.json --finetune 2024model.pt

因为我们是multitask，所以不需要后面再指出用哪个branch，因为在json中已经显式地指出我们要用哪几个branch。DeePMD的微调模式，目前只支持调整模型最后一层的参数，意味着DPA-2描述符部分将会共享预训练模型中的参数。这也是为什么描述符部分的参数一定要和预训练模型一致。如果报错，大概率就是DPA-2描述符参数有误，请仔细修改。