

## 4.4 保存和加载数据

(English Version)

DGL建议用户实现保存和加载数据的函数，将处理后的数据缓存在本地磁盘中。这样在多数情况下可以帮用户节省大量的数据处理时间。 DGL提供了4个函数让任务变得简单。

- `dgl.save_graphs()` 和 `dgl.load_graphs()` : 保存DGLGraph对象和标签到本地磁盘和从本地磁盘读取它们。
- `dgl.data.utils.save_info()` 和 `dgl.data.utils.load_info()` : 将数据集的有用信息(python dict对象)保存到本地磁盘和从本地磁盘读取它们。

下面的示例显示了如何保存和读取图和数据集信息的列表。

```
import os
from dgl import save_graphs, load_graphs
from dgl.data.utils import makedirs, save_info, load_info

def save(self):
    # 保存图和标签
    graph_path = os.path.join(self.save_path, self.mode + '_dgl_graph.bin')
    save_graphs(graph_path, self.graphs, {'labels': self.labels})
    # 在Python字典里保存其他信息
    info_path = os.path.join(self.save_path, self.mode + '_info.pkl')
    save_info(info_path, {'num_classes': self.num_classes})

def load(self):
    # 从目录 `self.save_path` 里读取处理过的数据
    graph_path = os.path.join(self.save_path, self.mode + '_dgl_graph.bin')
    self.graphs, label_dict = load_graphs(graph_path)
    self.labels = label_dict['labels']
    info_path = os.path.join(self.save_path, self.mode + '_info.pkl')
    self.num_classes = load_info(info_path)['num_classes']

def has_cache(self):
    # 检查在 `self.save_path` 里是否有处理过的数据文件
    graph_path = os.path.join(self.save_path, self.mode + '_dgl_graph.bin')
    info_path = os.path.join(self.save_path, self.mode + '_info.pkl')
    return os.path.exists(graph_path) and os.path.exists(info_path)
```

请注意：有些情况下不适合保存处理过的数据。例如，在内置数据集 `GDELTDataset` 中，处理过的数据比较大。所以这个时候，在 `__getitem__(idx)` 中处理每个数据实例是更高效的方法。