

# Github的SSH密钥设定

## 1. 检查网络连接

确保你的网络连接稳定。如果网络不稳定，大文件的上传可能会被中断或延迟。

## 2. 增加 Git 的传输缓冲区大小

大文件上传时可以增加 Git 的传输缓冲区大小，来改善上传速度：

```
1 | git config --global http.postBuffer 524288000
```

这会将缓冲区大小设置为 500MB。

## 3. 使用 SSH 进行上传

如果 HTTPS 上传较慢，可以尝试使用 SSH 进行上传：

1. 生成 SSH 密钥（如果尚未生成）：

```
1 | ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

2. 添加 SSH 密钥到 SSH 代理：

```
1 | eval "$(ssh-agent -s)"
2 | ssh-add ~/.ssh/id_rsa
```

3. 将 SSH 公钥添加到 GitHub：

- 复制公钥到剪贴板：

```
1 | pbcopy < ~/.ssh/id_rsa.pub
```

- 登录到 GitHub，进入 `Settings` -> `SSH and GPG keys` -> `New SSH key`，将公钥粘贴进去。

4. 更改远程仓库 URL 为 SSH：

```
1 | git remote set-url origin git@github.com:Republic1024/Automode_LoRA.git
```

5. 再次推送：

```
1 | git push -u origin main
```

## 4. 检查是否存在上传进度

如果上传过程中没有任何进展，可以尝试中断上传并重新开始：

```
1 # 中断当前上传
2 Ctrl + C
3
4 # 再次尝试推送
5 git push -u origin main
```

## 5. 使用 GitHub 大文件存储 (Git LFS)

如果你的项目包含大文件或二进制文件，建议使用 Git LFS (Large File Storage)：

### 1. 安装 Git LFS：

```
1 brew install git-lfs
2 git lfs install
```

### 2. 跟踪大文件：

```
1 git lfs track "*.bin" # 替换为实际文件类型
```

### 3. 添加并提交文件：

```
1 git add .gitattributes
2 git add .
3 git commit -m "Add large files"
```

### 4. 推送到远程仓库：

```
1 git push -u origin main
```

## 总结

在大文件上传过程中可能会遇到一些问题，以下是一些常见的解决方案：

### 1. 增加 Git 的传输缓冲区大小：

```
1 git config --global http.postBuffer 524288000
```

### 2. 使用 SSH 进行上传：

```
1 ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
2 eval "$(ssh-agent -s)"
3 ssh-add ~/.ssh/id_rsa
4 pbcopy < ~/.ssh/id_rsa.pub # 复制公钥到剪贴板
5 # 在 GitHub 添加公钥
6 git remote set-url origin git@github.com:Republic1024/Automode_LoRA.git
7 git push -u origin main
```

### 3. 使用 GitHub 大文件存储 (Git LFS)：

```
1 | brew install git-lfs
2 | git lfs install
3 | git lfs track "*.bin" # 替换为实际文件类型
4 | git add .gitattributes
5 | git add .
6 | git commit -m "Add large files"
7 | git push -u origin main
```

希望这些方法能够帮助你顺利将本地文件上传到 GitHub。