What ChatGPT o1 Can Do That 4o-mini Cannot

I recently encountered an issue with the `fontawesome5` package in the GitHub flow. The solution provided by 4o-mini (using TeX Live 2021 and `tlmgr install fontawesome5`) didn' t work, but 4o suggested a better approach (using TeX Live 2023 and still using `tlmgr` to install `fontawesome5`), which I' m currently trying. It didn' t completely fix the problem, but switching to TeX Live 2023 instead of 2021 has made a significant improvement.

4o-mini answer

The error you're seeing is related to `tlmgr` (TeX Live Manager) not being properly initialized in the user mode. Specifically, the message `tlmgr: user mode not initialized, please read the documentation!` indicates that `tlmgr` is not being set up correctly in the user environment.

Here' s a breakdown of how you can fix it:

Fixing the `tlmgr` User Mode Error

1. **Set `tlmgr` to User Mode Correctly**: You need to explicitly tell `tlmgr` to run in user mode and ensure it' s initialized properly. To do this in your script, use the following steps:
    - Make sure you are using `tlmgr` in user mode by setting the correct environment variable.
    - You can also initialize the user-specific TeX Live directory before using `tlmgr`.
2. **Initialize `tlmgr` in User Mode**: Before running the `tlmgr` commands, ensure that `tlmgr` is initialized with the following command:

    ```
    tlmgr init-usertree
    ```

    This will create the user-specific TeX Live tree in `~/texmf`. Once initialized, you can then run the `tlmgr update` and `install` commands.
3. **Modify Your GitHub Actions Workflow**: Update your GitHub Actions workflow file to include the initialization step. Here' s an example of how you can modify the workflow:

    ```
    - name: Set tlmgr to user mode
      run: |
        # Initialize the user mode TeX Live directory
        tlmgr init-usertree
        tlmgr update --self
        tlmgr install fontawesome5
    ```

Full Example:

If you' re running this as part of a larger GitHub Actions workflow, you can incorporate these changes into the relevant job section:

```
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Install TeX Live (xelatex)
        run: |
          sudo apt-get update
          sudo apt-get install -y texlive-xetex
          tlmgr init-usertree  # Initialize the TeX Live user tree
          tlmgr update --self  # Update tlmgr itself
          tlmgr install fontawesome5  # Install the desired package
```

Explanation of Commands:

- `tlmgr init-usertree`: This initializes the user-specific TeX Live tree (usually located in `~/texmf`).
- `tlmgr update --self`: Updates the `tlmgr` package itself to the latest version.
- `tlmgr install fontawesome5`: Installs the `fontawesome5` package.

Additional Notes:

- **Permissions**: Ensure that the user running the GitHub Actions workflow has permission to install and modify files in the TeX Live user directory.
- **Environment Variables**: If you' re working with custom paths, you may need to adjust `TEXMFSYSVAR` or `TEXMFHOME` to point to your specific directories.

By ensuring that `tlmgr` is initialized in user mode and setting it up correctly in your GitHub Actions workflow, the error should be resolved. Let me know if this works or if you encounter any other issues!

o1 Answer

I understand you' re encountering issues with `tlmgr` (TeX Live Manager) in your GitHub Actions workflow, specifically when trying to install the `fontawesome5` package. The error message:

```
TLPDB: not a directory, not loading: /home/runner/texmf
tlmgr: user mode not initialized, please read the documentation!
Error: Process completed with exit code 2.
```

indicates that `tlmgr` is unable to operate in “user mode” because the necessary directories and configurations are not set up correctly. This issue often arises when using TeX Live installations provided by package managers like `apt-get`, which can have limited functionality compared to a full TeX Live installation from TUG (TeX Users Group).

To resolve this, I recommend **installing TeX Live 2023 manually** within your GitHub Actions workflow. This approach ensures that `tlmgr` has full functionality and can install additional packages like `fontawesome5` without issues.

Step-by-Step Solution:

1. **Remove System TeX Live Installation:**
    - Avoid using `apt-get install texlive-xetex` as it provides a partial TeX Live installation with restricted `tlmgr` functionality.
2. **Manually Install TeX Live 2023:**
    - Download and install TeX Live directly from TUG.
3. **Configure PATH Environment Variable:**
    - Ensure that the TeX Live binaries are accessible by updating the PATH.
4. **Use `tlmgr` to Install Required Packages:**
    - Update `tlmgr` and install `fontawesome5`.
5. **Optimize Workflow with Caching (Optional):**
    - Cache the TeX Live installation to speed up future workflow runs.

Updated GitHub Actions Workflow:

Below is an updated version of your GitHub Actions workflow that incorporates the manual installation of TeX Live 2023 and correctly sets up `tlmgr` to install the `fontawesome5` package.

```yaml
name: Deploy Jekyll with GitHub Pages dependencies preinstalled

on:
  push:
    branches: ["master"]
  workflow_dispatch:

permissions:
  contents: read
  pages: write
  id-token: write
```

```
concurrency:
  group: "pages"
  cancel-in-progress: false

jobs:
  awesome-cv-copy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Make
        run: sudo apt-get update && sudo apt-get install -y make

      - name: Cache TeX Live
        uses: actions/cache@v3
        with:
          path: /usr/local/texlive/2023
          key: ${{ runner.os }}-texlive-2023
          restore-keys: |
            ${{ runner.os }}-texlive-

      - name: Install TeX Live 2023
        if: steps.cache-texlive.outputs.cache-hit != 'true'
        run: |
          # Install dependencies for TeX Live installer
          sudo apt-get update
          sudo apt-get install -y perl wget xz-utils

          # Download TeX Live installer
          wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
          tar -xzf install-tl-unx.tar.gz
          cd install-tl-*/

          # Create a texlive.profile for automated installation
          cat <<EOF > texlive.profile
          selected_scheme scheme-full
          TEXDIR /usr/local/texlive/2023
          TEXMFCONFIG ~/.texlive2023/texmf-config
          TEXMFHOME ~/texmf
          TEXMFLOCAL /usr/local/texlive/2023/texmf-local
          TEXMFSYSCONFIG /usr/local/texlive/2023/texmf-config
          TEXMFSYSVAR /usr/local/texlive/2023/texmf-var
          binary_x86_64-linux 1
          collection-basic 1
          collection-latex 1
          collection-fontsrecommended 1
          EOF

          # Run the installer with the profile
          sudo ./install-tl --profile=texlive.profile

      - name: Add TeX Live to PATH
        run: echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH

      - name: Update tlmgr and Install fontawesome5
        run: |
          tlmgr update --self
          tlmgr install fontawesome5

      - name: Run make awesome-cv-copy
        run: make awesome-cv-copy

  build:
    runs-on: ubuntu-latest
    needs: awesome-cv-copy
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Setup Python 3
```

```
      uses: actions/setup-python@v4
      with:
        python-version: "3.x"

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install python-frontmatter

    - name: Generate notes links
      run: |
        python scripts/generate_notes_link.py

    - name: Setup Pages
      uses: actions/configure-pages@v5

    - name: Build with Jekyll
      uses: actions/jekyll-build-pages@v1
      with:
        source: ./
        destination: ./_site

    - name: Upload artifact
      uses: actions/upload-pages-artifact@v3

deploy:
  environment:
    name: github-pages
    url: ${{ steps.deployment.outputs.page_url }}
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4
```

Detailed Explanation:

1. **Checkout Repository:**
    - Uses the `actions/checkout@v4` action to clone your repository.
2. **Set up Make:**
    - Installs `make` using `apt-get`, ensuring that it' s available for building your CV.
3. **Cache TeX Live:**
    - Utilizes `actions/cache@v3` to cache the TeX Live installation located at `/usr/local/texlive/2023`. This significantly reduces installation time for subsequent workflow runs.
4. **Install TeX Live 2023:**
    - **Conditional Execution:** The `if: steps.cache-texlive.outputs.cache-hit != 'true'` condition ensures that TeX Live is only installed if it' s not already cached.
    - **Install Dependencies:** Installs necessary dependencies like Perl, `wget`, and `xz-utils`.
    - **Download and Extract Installer:** Downloads the latest TeX Live installer from CTAN and extracts it.
    - **Create Installation Profile:** A `texlive.profile` is created to automate the installation process. This profile specifies:
        - **`selected_scheme scheme-full`:** Installs the full TeX Live distribution. You can adjust this to a minimal scheme if desired.
        - **Directories:** Specifies where TeX Live and its configurations are installed.
        - **Collections:** Selects essential collections like `basic`, `latex`, and `fontsrecommended`.
    - **Run Installer:** Executes the installer with the specified profile, installing TeX Live 2023 to `/usr/local/texlive/2023`.
5. **Add TeX Live to PATH:**
    - Updates the `PATH` environment variable by appending the TeX Live binary directory. This ensures that `tlmgr` and other TeX Live commands are accessible in subsequent steps.
6. **Update tlmgr and Install `fontawesome5`:**
    - **Update `tlmgr`:** Ensures that `tlmgr` itself is up to date.
    - **Install `fontawesome5`:** Uses `tlmgr` to install the `fontawesome5` package, resolving your initial issue.
7. **Run Make Command:**
    - Executes `make awesome-cv-copy` to build your CV using the installed TeX Live setup.
8. **Build and Deploy Jobs:**
    - These remain largely unchanged but ensure that the TeX Live installation is available throughout the workflow due to the updated PATH.

Additional Recommendations:

- **TeX Live Installation Time:**
  - **Caching:** The inclusion of the `actions/cache@v3` step helps mitigate the lengthy installation time by caching TeX Live after the first run. Ensure that the cache key (`${{ runner.os }}-texlive-2023`) accurately reflects the TeX Live version to avoid cache mismatches.
- **Customizing TeX Live Installation:**
  - If you don' t need the full TeX Live distribution, you can modify the `texlive.profile` to install only specific collections, reducing installation time and disk usage. For example, replace `scheme-full` with `scheme-basic` and specify only the necessary collections.
- **Verifying Installation:**
  - To ensure that TeX Live is installed correctly, you can add a verification step:

    ```
    - name: Verify TeX Live Installation
      run: |
        tex --version
        tlmgr --version
        kpsewhich fontawesome5.sty
    ```

    This step can help diagnose any issues by confirming that TeX Live commands are accessible and that `fontawesome5.sty` is installed.
- **Handling PATH Correctly:**
  - Using `echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH` ensures that the TeX Live binaries are added to the PATH for all subsequent steps in the job. This is crucial for `tlmgr` and LaTeX commands to work correctly.

Troubleshooting Tips:

1. **Cache Miss Issues:**
   - Ensure that the cache key accurately reflects the TeX Live version. If you update TeX Live or change the installation path, update the cache key accordingly.
2. **Installation Failures:**
   - Check the GitHub Actions logs for any errors during the TeX Live installation. Common issues include network problems when downloading the installer or insufficient permissions.
3. **Package Installation Errors:**
   - If `tlmgr` fails to install `fontawesome5`, verify that the package name is correct and that it' s available in the TeX Live repository.
4. **PATH Issues:**
   - Ensure that the PATH is updated correctly. You can add a step to print the PATH for debugging:

     ```
     - name: Print PATH
     ```

```
          run: echo $PATH
```

5. **Disk Space Limitations:**
   - GitHub Actions runners have limited disk space. A full TeX Live installation can consume significant space. If you encounter disk space issues, consider installing a minimal TeX Live setup.

Final Workflow Example with Verification:

Here's a more comprehensive workflow that includes verification steps to ensure everything is set up correctly:

```
name: Deploy Jekyll with GitHub Pages dependencies preinstalled

on:
  push:
    branches: ["master"]
  workflow_dispatch:

permissions:
  contents: read
  pages: write
  id-token: write

concurrency:
  group: "pages"
  cancel-in-progress: false

jobs:
  awesome-cv-copy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Make
        run: sudo apt-get update && sudo apt-get install -y make

      - name: Cache TeX Live
        uses: actions/cache@v3
        with:
```

```
    path: /usr/local/texlive/2023
    key: ${{ runner.os }}-texlive-2023
    restore-keys: |
      ${{ runner.os }}-texlive-

- name: Install TeX Live 2023
  if: steps.cache-texlive.outputs.cache-hit != 'true'
  run: |
    # Install dependencies for TeX Live installer
    sudo apt-get update
    sudo apt-get install -y perl wget xz-utils

    # Download TeX Live installer
    wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
    tar -xzf install-tl-unx.tar.gz
    cd install-tl-*/

    # Create a texlive.profile for automated installation
    cat <<EOF > texlive.profile
    selected_scheme scheme-full
    TEXDIR /usr/local/texlive/2023
    TEXMFCONFIG ~/.texlive2023/texmf-config
    TEXMFHOME ~/texmf
    TEXMFLOCAL /usr/local/texlive/2023/texmf-local
    TEXMFSYSCONFIG /usr/local/texlive/2023/texmf-config
    TEXMFSYSVAR /usr/local/texlive/2023/texmf-var
    binary_x86_64-linux 1
    collection-basic 1
    collection-latex 1
    collection-fontsrecommended 1
    EOF

    # Run the installer with the profile
    sudo ./install-tl --profile=texlive.profile

- name: Add TeX Live to PATH
```

```
      run: echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH

    - name: Verify TeX Live Installation
      run: |
        tex --version
        tlmgr --version
        kpsewhich fontawesome5.sty || echo "fontawesome5.sty not found"

    - name: Update tlmgr and Install fontawesome5
      run: |
        tlmgr update --self
        tlmgr install fontawesome5

    - name: Verify fontawesome5 Installation
      run: kpsewhich fontawesome5.sty

    - name: Run make awesome-cv-copy
      run: make awesome-cv-copy

build:
  runs-on: ubuntu-latest
  needs: awesome-cv-copy
  steps:
    - name: Checkout repository
      uses: actions/checkout@v4

    - name: Setup Python 3
      uses: actions/setup-python@v4
      with:
        python-version: "3.x"

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install python-frontmatter
    - name: Generate notes links
      run: |
        python scripts/generate_notes_link.py

    - name: Setup Pages
      uses: actions/configure-pages@v5

    - name: Build with Jekyll
      uses: actions/jekyll-build-pages@v1
      with:
        source: ./
        destination: ./_site

    - name: Upload artifact
      uses: actions/upload-pages-artifact@v3

deploy:
  environment:
    name: github-pages
    url: ${{ steps.deployment.outputs.page_url }}
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4
```

Key Enhancements:

1. **Verification Steps:**
   - **Verify TeX Live Installation:** Ensures that `tex`, `tlmgr`, and `fontawesome5.sty` are accessible.
   - **Verify `fontawesome5` Installation:** Confirms that the `fontawesome5.sty` file is installed correctly.
2. **Improved Caching:**
   - **Cache Key Accuracy:** The cache key is based on the OS and TeX Live version, ensuring that changes to TeX Live necessitate a new cache.
3. **Error Handling:**

- **Fallback Message:** The `kpsewhich fontawesome5.sty || echo "fontawesome5.sty not found"` command provides a fallback message if the package isn’t found, aiding in debugging.

Additional Tips:

- **Minimal TeX Live Installation:**
  - If the full TeX Live installation is too time-consuming or consumes too much space, consider installing a minimal scheme and adding only the necessary packages.
  - Update the `texlive.profile` accordingly:

    ```
    selected_scheme scheme-basic
    collection-latexrecommended 1
    collection-fontsrecommended 1
    collection-fontsextra 1
    collection-luatex 1
    ```
- **Parallelizing Jobs:**
  - Depending on your workflow’s complexity, you might consider parallelizing jobs or reusing steps across jobs to optimize runtime.
- **Handling TeX Live Updates:**
  - Regularly updating `tlmgr` and TeX Live ensures compatibility and access to the latest packages.

Conclusion:

By manually installing TeX Live 2023 within your GitHub Actions workflow and properly configuring the environment, you can overcome the limitations and errors associated with the system-installed TeX Live packages. This setup provides full `tlmgr` functionality, allowing you to install additional LaTeX packages like `fontawesome5` seamlessly.

If you continue to experience issues, please provide more detailed logs or specific error messages, and I’ll be happy to assist further!