Sure! Here's a detailed workflow that incorporates the relevant links, papers, and resources. This plan focuses on trying a few techniques at a time, measuring their impact, and iteratively improving your model for the AIMO Kaggle competition using the Agent Zero framework.

Overall Workflow Outline

1. **Initial Understanding and Setup**
    - **Weeks 1-2**: Deep dive into Agent Zero's core structure.
        - **Activities**:
            - Review repository code.
            - Watch instructional videos.
        - **Resources**:
            - Agent Zero GitHub Repository
            - Agent Zero Framework Introduction Video
            - Agent Zero YouTube Channel
    - **Establish Baseline Model**.
2. **Research and Prioritization**
    - Identify and prioritize techniques.
    - **Activities**:
        - Read and summarize relevant papers.
        - Discuss with team members.
3. **Implementation and Experimentation**
    - **Iterative Cycles** (Each cycle focuses on a new technique or combination).
        - **Design Controlled Experiments**.
        - **Implement Selected Techniques**.
        - **Test and Evaluate**.
4. **Evaluation and Iteration**
    - Analyze results.
    - Decide on next steps based on findings.
5. **Scaling and Integration**
    - Combine successful techniques.
    - Optimize and refine the model.
6. **Documentation and Communication**
    - Keep detailed records.
    - Regular team meetings.

Detailed Workflow

Weeks 1-2: Initial Understanding and Setup

Objective: Gain a deep understanding of the Agent Zero framework to effectively build upon it.

Activities:

1. **Review Agent Zero Repository Code**:
    - **Action**: Clone the repository and explore the codebase.
        - `git clone https://github.com/frdel/agent-zero.git`
    - **Focus Areas**:
        - Core architecture.
        - How agents are defined and interact.
        - Integration of tools within the framework.
2. **Watch Instructional Videos**:
    - **Agent Zero Framework Introduction**:
        - Agent Zero Framework Video
    - **Agent Zero YouTube Channel**:
        - Agent Zero Channel
    - **Best Practices**:
        - Take notes on key concepts.
        - List down any questions or areas of confusion for further research.
3. **Set Up Development Environment**:
    - **Action**: Configure your local environment to run Agent Zero.
        - Ensure compatibility with required Python versions and dependencies.
        - Install necessary libraries using `pip install -r requirements.txt`.
4. **Run Sample Agents**:
    - **Action**: Execute sample agents provided in the repository to understand practical implementations.
    - **Best Practices**:
        - Debug and step through code execution.
        - Modify parameters to see effects.

Resources:

- **Agent Zero GitHub Repository**: https://github.com/frdel/agent-zero
- **Agent Zero Framework Introduction Video**: https://www.youtube.com/watch?v=_Pionjv4hGc
- **Agent Zero YouTube Channel**: https://www.youtube.com/@AgentZeroFW

Week 3: Establish Baseline Model

Objective: Create a baseline model using Agent Zero without any advanced techniques.

Activities:

1. **Implement Baseline Agent**:
    - **Action**: Develop a simple agent capable of solving basic mathematical problems.

- **Best Practices**:
  - Keep the model simple to isolate the effects of future techniques.
  - Ensure the agent can interact with the competition data format.

2. **Initial Submission**:
   - **Action**: Submit the baseline model to the competition.
   - **Metrics**:
     - Record the submission score.
     - Note any feedback from the competition platform.

3. **Documentation**:
   - **Action**: Document the baseline model's architecture, parameters, and performance.
   - **Best Practices**:
     - Use clear and concise language.
     - Include any assumptions or limitations.

Week 4: Research and Prioritization

Objective: Identify which advanced techniques to implement first based on potential impact and feasibility.

Activities:

1. **Literature Review**:
   - **Action**: Read and summarize relevant research papers.
   - **Key Papers and Resources**:
     - **Neuro-Symbolic Methods**:
       - *Neural Theorem Provers* by Rocktäschel & Riedel (2017): Link
     - **Monte Carlo Tree Search (MCTS)**:
       - *Teaching Language Models to Reason with Search* by Yao et al. (2023): Link
     - **Chain-of-Thought Prompting (CoT)**:
       - *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models* by Wei et al. (2022): Link
     - **Self-Consistency in CoT**:
       - *Self-Consistency Improves Chain-of-Thought Reasoning in Language Models* by Wang et al. (2022): Link
     - **Tool-Integrated Reasoning**:
       - *Program-Aided Language Models (PAL)* by Gao et al. (2022): Link
     - **Reinforcement Learning for Theorem Proving**:
       - *Deep Reinforcement Learning for Symbolic Mathematics* by Lample & Charton (2019): Link
     - **Curriculum Learning with Synthetic Data**:
       - *Measuring Mathematical Problem Solving With the MATH Dataset* by Hendrycks et al. (2021): Link
     - **AlphaGeometry and AlphaProof**:

        - *Solving Olympiad Geometry Without Human Demonstrations* by Trinh et al. (2024): Link
        - **DeepMind Blog on AlphaGeometry**: Link
    - **Numina Team's Approach**:
        - *Winning the AI-Mathematical Olympiad Progress Prize* on Hugging Face: Link
- **Best Practices**:
    - Summarize each paper focusing on methodologies and results.
    - Identify techniques relevant to your project.

2. **Prioritize Techniques**:
    - **Action**: Rank techniques based on potential impact and implementation complexity.
    - **Criteria**:
        - Expected improvement in model performance.
        - Required resources and time.
        - Compatibility with Agent Zero.

3. **Team Discussion**:
    - **Action**: Meet with team members to decide on which techniques to implement first.
    - **Best Practices**:
        - Encourage open discussion.
        - Assign roles based on expertise.

Week 5: Implementing First Technique – Chain-of-Thought Prompting

Objective: Enhance the agent's reasoning by implementing CoT prompting.

Activities:

1. **Implement Chain-of-Thought (CoT) Prompting**:
    - **Action**: Modify the language model to generate intermediate reasoning steps.
    - **Guidelines**:
        - Use prompting techniques from the CoT paper by Wei et al. (2022).
        - Design prompts that encourage the model to think step-by-step.

2. **Internal Testing**:
    - **Action**: Evaluate the modified agent on a validation set.
    - **Metrics**:
        - Accuracy on validation problems.
        - Quality of generated reasoning steps.

3. **Submission and Evaluation**:
    - **Action**: Submit the updated model to the competition.
    - **Best Practices**:
        - Compare the new score with the baseline.
        - Analyze any changes in performance.

4. **Documentation**:
    - **Action**: Document changes made and results observed.
    - **Include**:
        - Examples of improved reasoning.
        - Challenges faced during implementation.

Week 6: Implementing Second Technique – Monte Carlo Tree Search (MCTS)

Objective: Improve decision-making and search efficiency using MCTS.

Activities:

1. **Integrate MCTS into Agent Zero**:
    - **Action**: Implement MCTS algorithms within the agent's framework.
    - **Resources**:
        - *Teaching Language Models to Reason with Search* by Yao et al. (2023): Link
        - MCTS Tutorials and Implementations:
            - **Wikipedia Overview**: Monte Carlo Tree Search
            - **Tutorial**: A Beginner's Guide to MCTS
    - **Best Practices**:
        - Start with a simple MCTS implementation.
        - Ensure integration does not disrupt existing functionalities.
2. **Test and Validate**:
    - **Action**: Assess the performance improvement on complex problems.
    - **Metrics**:
        - Solution accuracy.
        - Computational efficiency.
3. **Submission and Analysis**:
    - **Action**: Submit the MCTS-enhanced model.
    - **Best Practices**:
        - Record submission score.
        - Analyze whether MCTS provides a significant advantage.

Week 7: Combining CoT and MCTS

Objective: Explore the synergistic effects of combining CoT prompting with MCTS.

Activities:

1. **Integrate Techniques**:
    - **Action**: Modify the agent to use CoT prompting within the MCTS framework.
    - **Considerations**:
        - How reasoning steps influence tree exploration.
        - Balancing exploration vs. exploitation in the search.
2. **Implement Self-Consistency**:

- **Action**: Use self-consistency methods to improve reliability.
- **Resources**:
  - *Self-Consistency Improves Chain-of-Thought Reasoning in Language Models* by Wang et al. (2022): Link
- **Best Practices**:
  - Generate multiple reasoning paths.
  - Aggregate results to find the most consistent solution.

3. **Testing and Submission**:
   - **Action**: Evaluate the combined model and submit.
   - **Metrics**:
     - Improvement over individual techniques.
     - Analysis of where the combination excels or fails.

Week 8: Integrating External Tools

Objective: Equip the agent with external tools for symbolic computations.

Activities:

1. **Select and Integrate Tools**:
   - **Action**: Incorporate tools like SymPy or other mathematical libraries.
   - **Resources**:
     - **SymPy Documentation**: https://www.sympy.org/en/index.html
     - **Tool-Integrated Reasoning Paper**:
       - *Program-Aided Language Models (PAL)* by Gao et al. (2022): Link

2. **Modify Agent Architecture**:
   - **Action**: Allow the agent to call external tools during problem-solving.
   - **Best Practices**:
     - Ensure secure and efficient tool integration.
     - Handle errors gracefully.

3. **Test Complex Problem Solving**:
   - **Action**: Evaluate the agent's performance on problems requiring symbolic manipulation.
   - **Metrics**:
     - Accuracy on new problem types.
     - Speed of computation.

4. **Submission and Evaluation**:
   - **Action**: Submit the updated model and analyze results.
   - **Documentation**:
     - Detail the integration process.
     - Note any limitations encountered.

Week 9: Synthetic Data Generation and Fine-Tuning

Objective: Enhance the model's capabilities by training on a larger, more diverse dataset.

Activities:

1. **Generate Synthetic Data**:
    - **Action**: Use models like OpenAI's o1 to create additional reasoning datasets.
    - **Resources**:
        - OpenAI's GPT-4 Technical Report (2023): Link
        - **AlphaCode**: DeepMind's AlphaCode
    - **Best Practices**:
        - Ensure data quality and relevance.
        - Avoid overfitting to synthetic data.
2. **Fine-Tune Language Model**:
    - **Action**: Retrain the language model with the new data.
    - **Considerations**:
        - Monitor for any degradation in performance on real-world data.
        - Validate on a separate test set.
3. **Evaluate Impact**:
    - **Action**: Test the fine-tuned model and submit.
    - **Metrics**:
        - Improvement in problem-solving accuracy.
        - Ability to handle a wider variety of problems.

Weeks 10-12: Iteration and Scaling

Objective: Refine successful techniques and explore advanced methods.

Activities:

1. **Ablation Studies**:
    - **Action**: Systematically remove or modify components to understand their impact.
    - **Best Practices**:
        - Change one variable at a time.
        - Document findings meticulously.
2. **Hyperparameter Tuning**:
    - **Action**: Optimize model parameters for better performance.
    - **Tools**:
        - Use libraries like Optuna or Hyperopt.
        - Consider Bayesian optimization methods.
3. **Explore Advanced Techniques**:
    - **Reinforcement Learning for Theorem Proving**:
        - **Action**: Implement RL methods to train the agent.
        - **Resources**:

        - *Deep Reinforcement Learning for Symbolic Mathematics* by Lample & Charton (2019): Link
        - **AlphaProof Paper**: DeepMind's AlphaProof
    - **Curriculum Learning**:
        - **Action**: Organize training data to gradually increase difficulty.
        - **Resources**:
            - *Measuring Mathematical Problem Solving With the MATH Dataset* by Hendrycks et al. (2021): Link

4. **Combine Techniques**:
    - **Action**: Integrate reinforcement learning with previous methods.
    - **Best Practices**:
        - Ensure compatibility between techniques.
        - Monitor for unintended interactions.

Ongoing: Documentation and Communication

Objective: Maintain clear communication and thorough documentation throughout the project.

Activities:

1. **Regular Team Meetings**:
    - **Action**: Schedule weekly meetings to discuss progress and challenges.
    - **Best Practices**:
        - Prepare agendas beforehand.
        - Assign action items after meetings.

2. **Experiment Tracking**:
    - **Tools**:
        - Use MLflow or Weights & Biases for tracking.
    - **Action**: Log all experiments, parameters, and results.
    - **Benefits**:
        - Facilitates reproducibility.
        - Aids in identifying trends.

3. **Central Repository**:
    - **Action**: Use a shared platform like GitHub or Freedcamp for code and documents.
    - **Best Practices**:
        - Keep the repository organized.
        - Use clear commit messages and pull requests.

Additional Resources and Best Practices

- **Project Management Tools**:
    - **Freedcamp**: https://freedcamp.com/
    - **Trello**: https://trello.com/

    - **Asana**: https://asana.com/
- **Version Control**:
    - **Git Best Practices**:
        - Use branching strategies.
        - Regularly merge and resolve conflicts.
- **Data Management**:
    - **Organization**:
        - Keep raw and processed data separate.
        - Version data if it changes over time.
- **Collaboration and Communication**:
    - **Slack or Microsoft Teams** for instant communication.
    - **Shared Documentation Platforms**:
        - Google Docs or Notion for collaborative writing.

Final Thoughts

By following this detailed workflow and integrating the provided resources, you can methodically experiment with different techniques, measure their impact, and iteratively improve your model for the competition. Remember to balance the exploration of new methods with the depth of understanding and implementation quality.

Best Practices:

- **Time Management**: Allocate time wisely, focusing on high-impact activities.
- **Continuous Learning**: Stay updated with the latest research on platforms like arXiv and Papers with Code.
- **Team Collaboration**: Leverage the strengths of your team members and maintain open communication.

Good Luck with Your Competition!