Complete Research To-Do List - Aligned with Proposal

📋 Phase 1: Foundation Setup (June 27 - July 5)

1.1 Environment Setup

- [ ] Install MuJoCo physics engine
- [ ] Install gymnasium, stable-baselines3, dm-control
- [ ] Set up project structure as designed
- [ ] Create virtual environment
- [ ] Install and configure Weights & Biases (wandb)
- [ ] Test basic imports work

1.2 Get RealAnt Working

- [ ] Install RealAnt-RL from Ote Robotics (https://github.com/AaltoVision/realant-rl)
- [ ] Verify environment loads: `env = gym.make('RealAnt-v0')`
- [ ] Document observation space structure:
  - Joint positions [0:8]
  - Joint velocities [8:16]
  - Base orientation quaternion [16:20]
  - Base velocity [20:23]
  - Base angular velocity [23:26]
  - Contact sensors [26:28]
- [ ] Document action space: 8 continuous joint torques
- [ ] Verify 8 DOF (2 joints per leg: hip and ankle)
- [ ] Record video of random policy baseline

1.3 Define Success Metrics (from Section 4.1)

- [ ] **Success Rate**: Forward locomotion > 1.5m in 5 seconds
- [ ] **Cumulative Reward**: Sum of episode rewards
- [ ] **Recovery Time**: Time to resume walking after fault
- [ ] **Failure Rate**: % episodes with collapse/spin/stuck >2s
- [ ] **Goal velocity**: 0.5-1.0 m/s target
- [ ] **Episode length**: 500 timesteps
- [ ] Create evaluation script implementing all metrics

📋 Phase 2: PPO Baseline (July 6 - July 15)

2.1 Implement PPO Architecture (Section 3.2)

- [ ] Create policy network:
  - Input: 28-dimensional observation vector
  - Hidden layers: [64, 128] with ReLU activation
  - Output: 8-dimensional continuous actions
- [ ] Create value network (critic):
  - Same encoder as policy
  - Output: scalar value estimate
- [ ] Implement PPO loss with clipping (Equation 3.1)
- [ ] Use stable-baselines3 as base

2.2 Configure PPO Hyperparameters (Section 3.4)

- [ ] Learning rate: 3×10^{-4}
- [ ] Batch size: 2048
- [ ] Epochs per update: 10
- [ ] Clipping parameter (ε): 0.2
- [ ] Discount factor (γ): 0.99
- [ ] GAE parameter (λ): 0.95
- [ ] Create config file with these exact values

2.3 Design Reward Function

- [ ] Forward velocity reward (primary)
- [ ] Alive bonus: 0.1
- [ ] Control cost penalty: 0.01
- [ ] Implement in custom reward wrapper

2.4 Train and Evaluate Baseline

- [ ] Train for 1M steps initially
- [ ] Log to TensorBoard and W&B
- [ ] Save checkpoints every 50k steps
- [ ] Evaluate on 100 episodes
- [ ] Target: >90% success rate on clean environment
- [ ] Document baseline performance

📋 Phase 3: SR²L Implementation (July 16 - July 25)

3.1 Implement SR²L Loss (Section 3.2, Equation 3.2)

- [ ] Add smooth regularization term:

```
python

L_smooth = E[||π(s) - π(s + δ)||²]
where δ ~ N(0, σ²I)
```

- [ ] Set perturbation std (σ) for δ
- [ ] Implement combined loss (Equation 3.3):

```
python

L_total = L_PPO + λ * L_smooth
```

- [ ] Set λ = 0.01 (from paper)

3.2 Modify PPO Training Loop

- [ ] Create batch of perturbed observations
- [ ] Compute policy outputs for both clean and perturbed
- [ ] Calculate smoothness loss
- [ ] Add to PPO objective
- [ ] Log smooth_loss separately

3.3 Train PPO + SR²L

- [ ] Use same hyperparameters as baseline
- [ ] Train for same duration
- [ ] Monitor both PPO loss and smooth loss
- [ ] Verify smooth loss decreases

3.4 Evaluate Smoothness

- [ ] Compare action sequences between PPO and PPO+SR²L
- [ ] Measure action derivative/jerkiness
- [ ] Success rate should remain >90%
- [ ] Document smoothness improvements

📋 Phase 4: Domain Randomization Setup (July 26 - August 5)

4.1 Implement Fault Injection Wrapper (Section 3.3)

- [ ] Create `FaultInjectionWrapper(gym.Wrapper)`
- [ ] Implement actuator fault modes:
  - **Lock mode**: Use PD control to maintain position
    - Kp = 100.0 (proportional gain)

    - Kd = 10.0 (derivative gain)
- **Zero torque**: Set action to 0
- **Weak motor**: Multiply by 0.3 factor

☐ Joint selection logic:

- Random selection from 8 joints
- Option for coupled failures (both joints in leg)

4.2 Implement Sensor Noise (Section 3.3, Equation 3.4)

☐ Add Gaussian noise: $\tilde{s} = s + \varepsilon, \varepsilon \sim N(0, \sigma^2 I)$

☐ Configure noise levels:

- Position noise: σ = 0.05
- Velocity noise: σ = 0.1
- Orientation noise: σ = 0.02

☐ Apply noise per timestep

☐ Handle quaternion normalization

4.3 Create Curriculum Manager (Section 3.4)

☐ Implement 3-phase curriculum:

Phase 1: Warm-up (Epochs 0-200)

- No actuator faults
- Minimal sensor noise (σ = 0.01)
- Goal: Learn base locomotion

Phase 2: Isolated Faults (Epochs 200-600)

- Single joint dropout per episode
- Fault probability: 0.2
- Sensor noise: σ = 0.05
- Goal: Learn compensation

Phase 3: Full Randomization (Epochs 600+)

- Multiple joint faults (up to 3)
- Fault probability: 0.4
- Sensor noise: σ = 0.1

- Goal: Maximum robustness

4.4 Test Fault Injection

- [ ] Verify joints actually lock/fail
- [ ] Check sensor noise is applied
- [ ] Visualize robot with faults
- [ ] Log fault statistics

📋 Phase 5: PPO + DR Training (August 6 - August 15)

5.1 Integrate Components

- [ ] Wrap environment with fault injection
- [ ] Connect curriculum manager
- [ ] Ensure curriculum phases transition correctly
- [ ] Log current phase and fault stats

5.2 Extended Training

- [ ] Train for 10M steps (full curriculum)
- [ ] Monitor performance per phase
- [ ] Track success rate vs fault severity
- [ ] Save checkpoints at phase transitions

5.3 Ablation: PPO + SR²L (No Faults)

- [ ] Train with SR²L but no domain randomization
- [ ] Same 10M steps
- [ ] Evaluate robustness without fault training

📋 Phase 6: Full Method Training (August 16 - August 25)

6.1 PPO + DR + SR²L Combined

- [ ] Enable all components:
  - PPO base algorithm
  - SR²L smoothness ($\lambda = 0.01$)
  - Domain randomization
  - Curriculum learning
- [ ] Train for 10M steps
- [ ] Monitor all losses

6.2 Complete All Ablations

Ensure all 4 variants are trained:

- [ ] PPO only (baseline)
- [ ] PPO + SR²L
- [ ] PPO + DR
- [ ] PPO + DR + SR²L

6.3 Checkpoint Management

- [ ] Save best model from each variant
- [ ] Save at 1M, 5M, 10M steps
- [ ] Document training curves

📋 Phase 7: Evaluation (August 26 - September 5)

7.1 Implement Evaluation Protocol (Section 4.2)

Test each policy on 5 scenarios × 100 episodes each:

- [ ] **Clean environment** (no faults)
  - Target: Baseline maintains >95% success
- [ ] **Single joint locked** (random selection)
  - Target: >70% success rate
- [ ] **Multiple joint lock** (2-3 joints)
  - Target: >45% success rate
- [ ] **Sensor noise only**
  - Position/velocity/orientation noise
  - No actuator faults
- [ ] **Combined faults** (joints + noise)
  - Most challenging scenario
  - Measure graceful degradation

7.2 Statistical Analysis (Section 4.3)

- [ ] Compute mean ± std for all metrics
- [ ] Calculate 95% confidence intervals
- [ ] Run paired t-tests between methods
- [ ] Use chi-squared for success rates
- [ ] Create significance tables

7.3 Generate Visualizations

- [ ] Learning curves (reward over time)
- [ ] Success rate bar plots by condition
- [ ] Performance degradation curves
- [ ] Box plots for reward distributions
- [ ] Recovery time comparisons

📋 Phase 8: Analysis & Writing (September 6 - October 5)

8.1 Results Analysis

- [ ] Confirm hypothesis: Combined > Individual > Baseline
- [ ] Identify which component contributes most
- [ ] Document failure modes
- [ ] Analyze recovery strategies

8.2 Create Deliverables

- [ ] Results tables (LaTeX format)
- [ ] All required plots
- [ ] Video compilation showing:
  - Baseline walking
  - Single fault recovery
  - Multiple fault adaptation
  - Smooth vs jerky motions

8.3 Write Report Sections

Following proposal structure:

- [ ] Update methodology with actual implementation
- [ ] Write evaluation results
- [ ] Discuss findings
- [ ] Address limitations
- [ ] Future work recommendations

📋 Phase 9: Stretch Goals (If Time Permits)

9.1 Terrain-Aware Adaptation

- [ ] Add terrain variation (slopes, stairs)

- [ ] Train policy to choose paths based on damage

9.2 Vision Integration

- [ ] Add RGB/depth camera to observation
- [ ] Train terrain perception

9.3 Sim-to-Real Transfer

- [ ] Prepare policy for real RealAnt robot
- [ ] Test deployment pipeline

📋 Final Submission (October 6-14)

Final Checklist

- [ ] All code committed and documented
- [ ] Reproducibility instructions
- [ ] Final report formatted
- [ ] Videos and supplementary materials
- [ ] Submit by October 14

🎯 Key Milestones & Success Criteria

1. **Baseline Walking**: PPO achieves >90% success at 0.5+ m/s
2. **Smooth Motion**: SR²L shows measurably smoother actions
3. **Single Fault Robustness**: >70% success with one failed joint
4. **Multi-Fault Robustness**: >45% success with 2-3 failed joints
5. **Combined Method Best**: PPO+DR+SR²L outperforms all ablations

📊 Progress Tracking

Track daily progress with:

```
Date: YYYY-MM-DD
Completed: [List items]
Issues: [Any blockers]
Tomorrow: [Next tasks]
Training: [Current experiment status]
```

This comprehensive list now includes every technical detail from your proposal!