**Mechanical and Electrical Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **AIR** | **PRO** | **EDU** |
| **Dimension (Standing)** | 70 x 31 x 40 cm | 70 x 31 x 40 cm | 70 x 31 x 40 cm |
| **Dimension (Crouching)** | 76 x 31 x 20 cm | 76 x 31 x 20 cm | 76 x 31 x 20 cm |
| **Weight (with battery)** | About 15 kg | About 15 kg | About 15 kg |
| **Material** | Aluminium alloy + High strength engineering plastic | Aluminium alloy + High strength engineering plastic | Aluminium alloy + High strength engineering plastic |
| **Voltage** | 28V ~ 33.6V | 28V ~ 33.6V | 28V ~ 33.6V |
| **Peaking Capacity** | About 3000W | About 3000W | About 3000W |

**Performance:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **AIR** | **PRO** | **EDU** |
| **Payload** | ~8 kg (MAX ~ 10 kg) | ~7 kg (MAX ~ 10 kg) | ~8 kg (MAX ~ 12 kg) |
| **Speed** | 0 ~ 2.5 m/s | 0 ~ 3.5 m/s | 0 ~ 3.7 m/s |
| **Max Climb Drop Height** | About 15 cm | About 16 cm | About 16 cm |
| **Max Climb Angle** | 30° | 40° | 40° |
| **Basic Computing Power** | 8-core High-performance CPU | 8-core High-performance CPU | 8-core High-performance CPU |

**Knee Joint:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **AIR** | **PRO** | **EDU** |
| **Max Torque** | About 45 N-m | About 45 N-m | About 45 N-m |
| **Motor Type** | Aluminium knee joint motor | Aluminium knee joint motor | Aluminium knee joint motor |
| **Number of Sets** | 12 sets | 12 sets | 12 sets |
| **Range of Motion** | Body: -48° ~ 48°, Thigh: -200° ~ 90°, Shank: -156° ~ -48° | Body: -48° ~ 48°, Thigh: -200° ~ 90°, Shank: -156° ~ -48° | Body: -48° ~ 48°, Thigh: -200° ~ 90°, Shank: -156° ~ -48° |
| **Intra-joint Circuit (Knee)** | Yes | Yes | Yes |
| **Joint Cooling System** | Joint Heat Pipe Cooler | Joint Heat Pipe Cooler | Joint Heat Pipe Cooler |





**Upgrades to Bring the Pro to EDU Level**

To convert a Go2 Pro into EDU-equivalent, you’d need:

1. NVIDIA Jetson Orin Module  
   Splash in either 40 TOPS (standard EDU) or 100 TOPS (for mid‑360/Hesai EDU)
2. Foot-end force sensors  
   Add force-sensitive pads to the legs to match EDU sensor suite
3. SDK/Firmware Updates  
   Flash EDU firmware or custom SDK tools to unlock DDS-based remote control and
4. Optional: LIDAR Enhancements  
   Upgrade to mid‑360 3D LIDAR or Hesai XT16 if aiming for EDU Plus specs
5. Add Servo Arm & Accessories  
   If you need robotic arm support, depth cameras, docking station, etc. commonly offered as EDU add-ons

If your goal is full SDK access, onboard AI compute, advanced sensors, and plug-and-play educational capabilities, here’s the roadmap:

1. Install Jetson Orin (40 or 100 TOPS)
2. Integrate foot-end force sensors
3. Flash EDU-grade firmware / run official SDK
4. (Optional) Add 3D LIDAR and servo arm for education/research expansion

1. Install NVIDIA Jetson Orin Module

Jetson Orin Nano (40 TOPS) or Orin NX 16 GB (100 TOPS)

Steps:

1. Power off the robot and remove the top cover where the AI module sits.
2. Mount the Jetson board in the docking area, secure it with screws.
3. Connect power via the 16–60 V DC input, Ethernet, and USB cables (as specified in the Go2 docs)
4. Reattach the cover, then power on.
5. Flash JetPack OS (Ubuntu + CUDA) onto the Orin:
   * Boot Orin from SD or USB with flashed JetPack image.
   * Connect via HDMI/USB or SSH.
6. Verify with tegrastats or jtop that Orin is running correctly. Then integrate docking Ethernet/Wi‑Fi into robot network.

2. Foot‑End Force Sensors

What you need

Go2 EDU foot-end sensor modules (force-sensitive).

Wiring harness and firmware that supports them.

Steps:

1. Power off robot. Mount sensor pads on each foot, aligning connectors.
2. Route wiring through leg channels to main control board.
3. Secure harness and confirm connector interfaces.
4. Power on and update firmware to detect foot sensors.
5. Verify installation by running foot‑force diagnostics via Go2 App or control interface.

3. Flash EDU‑Grade Firmware & SDK

* Access to EDU firmware image.
* Unitree Go2 SDK (C++/Python or DDS) from Unitree docs

Connect robot to Wi‑Fi or docking station’s 192.168.123.18 (user: Unitree, pass: 123)

Download and flash EDU firmware via OTA or local upload (advanced tutorial on Unitree site)

Restart robot; confirm boot version indicates EDU mode.

|  |  |  |
| --- | --- | --- |
| **Upgrade Step** | **Needed Hardware** | **Software Task** |
| Install Jetson Orin | Orin module + mounts | JetPack install & integration |
| Add Foot Sensors | Force sensor pads + harness | Firmware update & diagnostics |
| Flash EDU Firmware & SDK | EDU firmware & SDK package | OTA flash, SDK install, demo programs |
| Optional: LiDAR & Arm | MID‑360 / XT16 LiDAR, Arm D1 | Mount, connect, ROS2 driver & test scripts |

**Component Analysis:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Details** | **Analysis** |
| **Powerful Computing Core** | - Motion Controller - High-performance ARM processor - NVIDIA Jetson Orin NX/Nano - Improved AI algorithm | Strong AI performance (up to 100 TOPS with Orin NX)  Supports real-time inference and robotics control  Needs configuration to unlock full SDK (for secondary development) |
| **Smart Battery** | - 800Wh (standard), 15000mAh - Smart BMS for overcharge/short protection | Excellent battery capacity for long endurance  Built-in protection circuitry  May not be hot-swappable like some research platforms |
| **Foot Force Sensor (EDU only)** | - Real-time feedback of foot pressure | **Not available on Go2 Pro** by default  Needed for terrain awareness, stable walking, adaptive locomotion  Can be added via aftermarket sensors & firmware upgrade |
| **Front Camera** | - 1280×720 resolution, FOV 120° - Ultra-wide-angle lens | Suitable for vision-based SLAM, obstacle detection  RGB only – for depth, external modules (like stereo or RealSense) needed |
| **Tracking Module** | - Supports auto or remote-controlled following | Built-in "Side-follow" ISS 2.0 using 4D LiDAR and AI vision  Effective for user interaction, filming, or autonomous movement |
| **Intercom Microphone** | - Voice scenario interaction | Useful for HMI (Human-Machine Interface)  SDK needed to access custom commands or NLP |
| **Front Light** | - Bright LED | Useful for low-light navigation, user visibility |
| **4D LiDAR L1** | - 360° omni-directional LiDAR - Small blind spot - Aids SLAM & obstacle avoidance | Major strength: L1 enables indoor/outdoor 3D navigation  Higher-end feature typically found in EDU-level robots |
| **12 Precision Joint Motors** | - Aluminum alloy - High-speed, durable - Visual expression capable | Allows smooth, agile movement (running, jumping)  Precision torque and speed control  High loads may require additional cooling or monitoring |
| **Speaker** | - For music playback or user interaction | Adds personality, useful for social interaction projects  Low-priority for research applications |

**Jailbreaking LLM-Controlled Robots**

**1. Overview:** The paper introduces **RoboPAIR**, the first algorithm designed to jailbreak large language model (LLM)-controlled robots, demonstrating vulnerabilities in robotic systems that could lead to harmful physical actions. The study evaluates three attack scenarios (white-box, gray-box, and black-box) across three commercial/academic robotic systems:

* **NVIDIA Dolphins** (self-driving LLM, white-box)
* **Clearpath Robotics Jackal UGV** (GPT-4o planner, gray-box)
* **Unitree Robotics Go2** (GPT-3.5-integrated robot dog, black-box)

The key finding is that **RoboPAIR achieves near 100% attack success rates (ASR)**, bypassing safety guardrails to elicit dangerous behaviors like bomb detonation, human collisions, and covert surveillance.

**2. Key Contributions: First Robotic Jailbreaking Algorithm**: Adapts the **PAIR** (Prompt Automatic Iterative Refinement) chatbot jailbreaking method to robotics by:

* + Incorporating robot-specific system prompts and APIs.
  + Adding a **syntax checker** to ensure generated code is executable.

1. **Three Threat Models**:
   * **White-box**: Full access to the LLM (e.g., NVIDIA Dolphins).
   * **Gray-box**: Partial access (e.g., Clearpath Jackal with GPT-4o).
   * **Black-box**: Query-only access (e.g., Unitree Go2).
2. **Real-World Demonstrations**:
   * Successfully jailbreaks deployed commercial robots (e.g., Unitree Go2).
   * Introduces **three datasets** of harmful robotic actions for benchmarking.
3. **Implications for AI Safety**:
   * Reveals risks beyond text generation (physical harm).
   * Highlights the need for **context-dependent alignment** in robotics.

**3. Methodology:**

* **RoboPAIR Workflow**:
  1. **Attacker LLM**: Generates adversarial prompts.
  2. **Target LLM (Robot)**: Executes the prompt.
  3. **Judge LLM**: Scores the harmfulness of the response (1–10).
  4. **Syntax Checker**: Validates API compatibility of generated code.
* **Baselines Compared**:
  1. Direct prompts (low ASR).
  2. In-context jailbreaks (high ASR).
  3. Template-based attacks (high ASR).
  4. Original PAIR (lower ASR due to lack of robotics focus).

**4. Key Results:**

|  |  |  |
| --- | --- | --- |
| **Robot System** | **Threat Model** | **Attack Success Rate (ASR)** |
| NVIDIA Dolphins | White-box | 100% (RoboPAIR) |
| Clearpath Jackal | Gray-box | 100% (RoboPAIR) |
| Unitree Go2 | Black-box | 100% (RoboPAIR) |

* **Unitree Go2**:
  + **API Jailbreaks**: Achieved 100% ASR by composing existing motion primitives (e.g., avoid\_obstacle(switch="off") to disable safety features).
  + **System Prompt Extraction**: First successful extraction of a commercial robot’s proprietary prompt.

**5. Discussion & Implications:**

1. **Context-Dependent Alignment**:
   * Unlike chatbots, robotic actions are **situational** (e.g., "walk forward" is harmless unless a human is in the path).
   * Current LLM alignment fails to account for environmental context.
2. **Physical Multi-Modal Risks**:
   * Robots combine **text, vision, and actuation**, expanding attack surfaces.
   * Jailbreaks can cause **real-world damage** (e.g., collisions, surveillance).
3. **Defense Challenges**:
   * Filter-based defenses (e.g., for chatbots) are inadequate for robotics.
   * Proposed solutions: **Physical safety constraints**, runtime monitoring, and adversarial training.

**SDK Development for Unitree PRO Model: Limitations and Guidelines:**

To develop the SDK for the PRO model of the Unitree humanoid robot, you can follow the specific guidelines and resources provided by Unitree. The PRO model does not support secondary development like the EDU version, which allows for extensive programming and customization. However, you can refer to the Motor SDK Development Guide provided by Unitree to understand the complete development process, from obtaining the SDK to burning and verifying code functions.

Here are the steps you can take to develop the SDK for the PRO model:

**Obtain the Motor SDK Development Guide from the provided file path: https://support.unitree.com/home/en/Motor\_SDK\_Dev\_Guide/Development\_Guide**

Follow the instructions in the guide to set up your development environment, configure parameters, and understand the SDK functionalities specific to the PRO model.

Utilize the resources and references in the guide to create your own applications tailored to the capabilities of the PRO model.

Ensure to adhere to any restrictions or limitations mentioned for the PRO model in terms of customization and programming.

While the PRO model may have restrictions compared to the EDU version, by following the Motor SDK Development Guide and leveraging the available resources, you can still develop applications and integrations for the PRO model of the Unitree humanoid robot. [For reference only]

The PRO model of Unitree robots does not support SDK for development like the EDU model does. The EDU model is specifically designed for educational purposes and research, enabling developers to customize and create applications using the available SDK. On the other hand, the PRO model focuses more on performance and features without the flexibility for secondary development through an SDK. Therefore, users of the PRO model will not have the same level of customization options and development opportunities that are available to EDU model users. If development and customization are primary requirements, the EDU model would be the better choice. [For reference only]