

NVIDIA RTX PRO 5000 BLACKWELL 48GB GDDR7



Cena celkem:	135 639 Kč (bez DPH: 112 098 Kč)
Běžná cena:	149 203 Kč
Ušetříte:	13 564 Kč
Kód zboží:	VGANVD1010
Part No.:	900-5G153-2550-000
Záruka:	36 měs.
Stav:	Nové zboží

Popis

NVIDIA RTX PRO 5000 Blackwell - pro specialisty všeho druhu

S **profesionální grafickou kartou NVIDIA RTX PRO 5000 Blackwell** zvládnete každý projekt s lehkostí. Je navržena pro využití v pracovních stanicích. Svým výkonem si poradí s provozem náročného grafického softwaru, AI, vizualizacemi, datovými analýzami, tvorbou obsahu, renderováním atd. Moderní **architektura Blackwell** přináší enormní výpočetní výkon v podobě **48 GB paměti GDDR7**, takže veškeré projekty zvládne bez sebemenšího zpomalení nebo zádrhelu. Nechybí ani **RTX platforma** s technologií **Ray Tracingu** nebo optimalizace pro stabilní FPS snímky v rámci **DLSS4**.



Grafická karta NVIDIA RTX PRO 5000 Blackwell je ideální pro inženýry, grafiky a další profesionály, kteří pracují s umělou inteligencí, tvoří, generují data a výpočty na pracovní i vědecké úrovni. Své uplatnění najde také v multimediálních systémech, které se zaměřují na kreativní tvorbu ve studiích a pracují s vysoce zátěžovým softwarem, ale potřebují s daty pracovat komplexně a bez omezení. Podpora **Multi-Instance GPU** umožní třídění výkonu mezi více pracovními procesy. Rozhraní **PCIe 5.0** a **DisplayPort 2.1b** zajistí extrémně rychlé datové přenosy, respektive podporu **až 16K rozlišení**.

```
4 - f rudimentary_chat_local():
5   # Load GPT-2 model and tokenizer
6   tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
7   model = GPT2LMHeadModel.from_pretrained("gpt2")
8
9   # Move model to GPU if available
10  device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
11  model.to(device)
12
13  print("Hello! I'm a basic GPT-2-based AI. Type 'quit' or 'exit' to end the chat.")
14
15  chat_history = ""
16
17  while True:
18    user_input = input("You: ")
19    if user_input.lower() in ["quit", "exit"]:
20      print("GPT-2-Chat: Goodbye!")
21      break
22
23    # Append user input to the chat history.
24    chat_history += f"User: {user_input}\nAI:"
25
26  # Print the chat history.
27  print(chat_history)
28
29  # End the function.
30  return chat_history
```

Ln: 51, Col: 1 pip install Package name ⬇

```
ging.getLogger(__name__)
ent:
nt for connecting the the lanchain-esque service."""
t_(self, server_url: str, model_name: str) -> None:
italize the client."""
server_url = server_url
_model_name = model_name
default_model = "local"
```

