Financira Evropska unija
NextGenerationEU

DIGITOP

NAČRT ZA OKREVANJE IN ODPORNOST

Program

Digitalna transformacija robotiziranih tovarn prihodnosti

DIGITOP

Poročilo D1.4

Metode za prenos veščin avtonomnega robota

Projekt/RRP	D1.4
Vrsta dokumenta	Poročilo o rezultatih
Verzija dokumenta	Končna
Sodelujoči partnerji	IJS-E1, IJS-E8
Avtorji	Andrej Gams, Boris Kuster, Aljaž Osojnik, Fatima Aziz, Nikola Marić, Boshko Koloski, Aleš Ude
Datum priprave	30.11.2025

Kazalo

1 Uvod

Prenos robotskih veščin med okolji oz. domenami, ki se razlikujejo po dinamiki, senzoričnih pogojih ali fizičnih značilnostih, ostaja temeljni raziskovalni problem v robotiki. Hkrati ta problem ni omejen samo na nestrukturirana okolja izven tovarn, kot so npr. domovi, temveč se izkazuje tudi v industrijskih alipkacijah. Vsako spremembo v proizvodnji ali izvedba nalog podobmnim tistim, ki se že izvajajo, lahko obravnavamo od začetka ali pa uporabimo že pridboljena znanja. Še posebej pomembno je vprašanje, ali se lahko proces oz. delovanje prilagaja avtonomno.

Tradicionalni pristopi, kot so učenje s posnemanjem [17], sbodbujevano učenje [7] ter celovito učenje od konca do konca (end-to-end) [15], lahko zahtevajo velike količine demonstracij, kar v realnih aplikacijah povzroča stroške, motnje pri delovanju proizvodnih procesov ter povečano kompleksnost podatkovne infrastrukture.

Napredni multimodalni modeli, kot so Flamingo, GPT-4o in drugi slikovno-jezikovni modeli (an. vision-language model, VLM) [10], spreminjajo paradigmo s tem, da omogočajo prenos znanja z minimalnim dodatnim učenjem. Njihova sposobnost semantične interpretacije slik, videov in besedil predstavlja ključno prednost pri posploševanju nalog, vendar jih primanjkljaj domensko specifičnih podatkov pogosto omejuje pri natančni prilagoditvi. Zato se v ospredje postavljajo hibridne tehnike, ki kombinirajo predznanje velikih modelov z majhnimi, selektivno izbranimi nabori podatkov.

Pristop z uporabo zunanje podatkovne baze za dopolitev (an. retrieval-augmeneted generation, RAG) je med temi tehnikami posebej pomemben zaradi svoje modularnosti in učinkovitosti. Hkrati pa se v sodobni literaturi kot ključni pristopi za prenos veščin pojavljajo še:

- meta-učenje, ki omogoča, da se modeli naučijo hitrega prilagajanja novim nalogam iz minimalnega števila primerov [2];
- prenos simulacija-realnost (an. sim-to-real), ki zmanjšuje odvisnost od realnih podatkov z uporabo naprednih fizikalnih simulacij [14];
- učenje iz nekaj primerov (an. few-shot learning), kjer se generalizacija doseže z izredno omejenim številom demonstracij [5];
- učenje latentnih reprezentacij, ki omogoča robustno primerjanje nalog in prenos strategij med okolji [11].

Konvergenca teh metod vodi v sistematično zmanjšanje podatkovnih zahtev pri uvajanju robotskih sistemov v nova okolja, hkrati pa odpira pomembna raziskovalna vprašanja, kot so dolgoročna stabilnost prenosa, zanesljivost v robnih primerih ter razvoj standardiziranih merilnikov za ocenjevanje sposobnosti generalizacije. S kombiniranjem VLM-jev, inteligentnih mehanizmov iskanja in naprednih tehnik prilagajanja se vzpostavlja nova generacija modelov, ki omogočajo robusten, skalabilen in učinkovit prenos znanja; s tem izboljšujemo avtonomnost in zanesljivost sodobnih robotskih sistemov.

1.1 Napovedovanje dejanj

Napovedovanje dejanj je ključno za delovanje robotskih sistemov v negotovih, nelinearnih in dinamično spreminjajočih se okoljih. Nedavni razvoj velikih slikovno-jezikovnih modelov (VLM-jev),

ki temeljijo na obsežnih naborih splošnega znanja, nakazuje nov paradigmatski premik v smeri univerzalnejših metod robotike. Zaradi svoje globoke semantične reprezentacije multimodalnih podatkov se VLM-ji vse bolj uporabljajo za kompleksne naloge predvidevanja dejanj, razumevanja prizorov in inferenc, ki presegajo tradicionalne, podatkovno intenzivne pristope. Vendarle, pa njihova nizka učinkovitost pri neposrednem prilagajanju na domensko specifične aplikacije ostaja ključna omejitev, saj so podatkovni nabori v robotiki pogosto premajhni, heterogeni ali dragi za zajem.

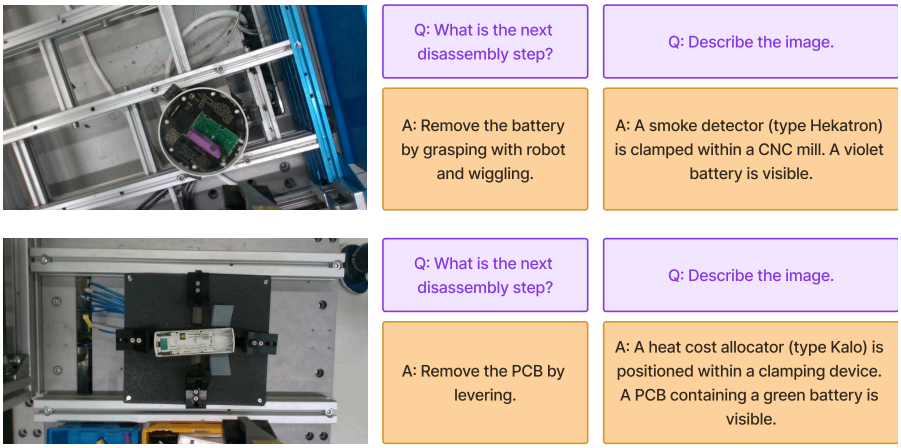
Eden najperspektivnejših pristopov za preseganje teh omejitev je uporaba pristopa RAG, ki temelji na dinamičnem vključevanju relevantnih primerov iz lokalnih baz znanja v vhodno predstavitev modela. Ta tehnika omogoča kontekstualno prilagoditev modela brez potrebe po klasičnem prilagajanju, saj VLM prek RAG dostopa do dodatnih specifičnih informacij za posamezno robotsko nalogo. V spodaj predstavljenem novem pristopu je izhodiščni RAG razširjen z več nadgradnjami: iterativnim izboljševanjem baze znanja z uporabo samega modela, optimizacijo izbranih regij slike, prilagodljivim razširjanjem iskalnega prostora ter ponovnim vrednotenjem pridobljenih primerov. Eksperimentalni rezultati evalvacije na področju predvidevanja robotskih dejanj reciklaže e-odpadkov kažejo 45-odstotno absolutno izboljšanje natančnosti v primerjavi z neposrednimi (zero-shot) napovedmi ter 20-odstotno izboljšanje glede na osnovni RAG.

2 Izboljšanje pristopa RAG za napovedovanje robotskih dejanj z uporabo VLM

Nedavno so veliki slikovno-jezikovni modeli prejeli več pozornosti na področju robotike zaradi njihovega obsežnega vgrajenega poznavanja sveta, ki omogoča upoštevanje splošnega znanja. To je težko vključiti ali pa sploh ni na voljo v klasičnih načrtovalskih domenah, npr. v jeziku Planning Definition Domain Language (PDDL) [4]. VLM-i imajo tudi zmožnost obdelave ene ali več slik ter sklepanje o objektih in njihovih odnosih [9]. Ključna prednost VLM-ov (in LLM-ov) je možnost dodajanja ustreznih primerov v strategijo pozivanja, kar je znano kot učni pristop s primeri (few-shot) oziroma učenje v kontekstu [1]. Čeprav lahko te ustrezne primere za vsako nalogo določi programer, bolj splošen pristop vključuje njihovo dinamično pridobivanje iz podatkovne baze na podlagi podobnosti med trenutnimi vhodi in elementi v podatkovni bazi. Ta pristop je znan kot generiranje, izboljšano s priklicem (Retrieval-Augmented Generation, RAG) [19] in omogoča izboljšanje delovanja VLM-ov brez računsko zahtevnega prilagajanja (fine-tuning), hkrati pa se izogne možnosti katastrofalnega pozabljanja in halucinacij [18].

Ideja RAG je imeti lokalno podatkovno bazo podatkovnih točk, pri čemer lahko vsaka podatkovna točka vsebuje sliko, besedilo in druge modalnosti ali njihovo kombinacijo, kot je prikazano na sliki 1.

Te podatkovne točke se nato vgradijo v vektorski prostor s pomočjo modalnostno specifičnega vgraditvenega (embedding) modela, ki je običajno bistveno manjši od VLM [16]. Ko se od odvodnega modela, npr. VLM, zahteva izvedbo inferenc na novi podatkovni točki, se elementi vhodne podatkovne točke, npr. vhodna slika, vgradijo z vgraditvenim modelom, nato pa se iz podatkovne baze pridobijo ustrezni primeri (slika in pripadajoče besedilo) ter dodajo v poziv (prompt) VLM. Ta pristop poveča natančnost napovedi VLM-jev [13]. Zlasti pri manjših podatkovnih zbirkah je lahko RAG učinkovitejši od fine-tuninga [8]. Nekatere pomembne prednosti uporabe sistema RAG so, da dodajanje podatkovnih točk ni programersko zahtevno, da je mogoče izboljšati zmogljivost



Slika 1: Primer dveh RAG podatkovnih točk, kjer ima vsaka sliko in tekst: par vprašanje in odgovor.

odvodnega modela, podatkovna baza pa se lahko po potrebi uporabi tudi kot podatkovna zbirka za fine-tuning.

V osnovnem pristopu RAG se iz podatkovne baze pridobi le najpodobnejša podatkovna točka in doda v VLM poziv. To ima več možnih slabosti:

1. Podatkovna baza morda ne vsebuje ustreznih primerov, vključevanje nerelevantnih podatkovnih točk pa lahko zmanjša natančnost napovedi.

2. Najbolj podobna podatkovna točka v vektorskem vgraditvenem prostoru morda ni najbolj reprezentativna, saj vgraditveni modeli niso popolnoma natančni.

3. Vhodna poizvedba lahko vsebuje moteče elemente, npr. sliko, kjer relevanten objekt zavzema le majhen del vhodne slike, kar zmanjša kakovost vgraditve.

4. Pri iskanju na osnovi besedila imajo vgraditveni modeli lahko težave z nestrukturiranim ali daljšim besedilom.

Da bi omilili te težave, smo razvili VLM-voden RAG, kjer je VLM uporabljen za izboljšanje (obrezovanje) vhodne slikovne poizvedbe in slik v podatkovni bazi RAG, za ponovno ovrednotenje pridobljenih rezultatov ter po potrebi za razširitev iskanja RAG, dokler VLM ne določi, da je bila najdena ustrezna podatkovna točka, ali dokler ni presežena vnaprej določena časovna omejitev. Ta postopek ustvari informativen primer, ki ga lahko uporabijo odvodni modeli. V kontekstu robotike, zlasti pri dinamičnih aplikacijah, kot je recikliranje, lahko to izboljša delovanje VLM-jev pri napovedovanju naslednjih dejanj in podobnih nalogah. Učinkovitost pristopa smo prikazali pri napovedovanju dejanj v nalogi razstavljanja odpadne elektronike. Takšni pristopi, ki temeljijo na umetni inteligenci, lahko bistveno zmanjšajo programersko delo, potrebno za prilagoditev robotskih delovnih celic novim aplikacijam [3].

2.1 Metodologija

Napovedovanje dejanj v robotiki predstavlja ključen izziv pri omogočanju večje avtomatizacije robotskih delovnih celic ter zmanjševanju programerskega dela pri njihovem prilagajanju novim

nalogam, kar na koncu robotom omogoča izvajanje bolj raznolikega in obsežnega nabora opravil. Tradicionalni pristopi temeljijo na nevronskih modelih, kot so večplastni perceptroni (MLP), rekurentne nevronske mreže (RNN) in LSTM-mreže [12]. Čeprav so ti modeli učinkoviti v specifičnih primerih, pogosto trpijo zaradi omejene sposobnosti posploševanja in zahtevajo učenje, prilagojeno posamezni nalogi.

V zadnjem času so VLM-ji na tem področju pokazali izjemne rezultate, saj predstavljajo nov pristop k napovedovanju naslednjih dejanj v robotskih delovnih celicah zaradi svoje sposobnosti generalizacije, velikih predtreniranih podatkovnih zbirk in zmožnosti interpretacije vizualnih vhodov [10]. Vendar pa so VLM-ji redko trenirani na podatkih, ki so podobni tistim pri napovedovanju robotskih dejanj. Njihovo prilagajanje (fine-tuning) ostaja zahtevno zaradi majhne velikosti razpoložljivih podatkovnih zbirk za napovedovanje dejanj ter velikega računalniškega stroška treniranja večjih modelov, ki praviloma prekašajo manjše [6].

Za izboljšanje natančnosti napovedovanja dejanj brez fine-tuninga je ena izmed praktičnih rešitev uporaba lokalne podatkovne baze primerov prek RAG. Ta pristop omogoča modelu, da med inferenco prikliče ustrezne primere in se nanje sklicuje, s čimer zapolni vrzel med splošnim znanjem VLM-ov in domeno specifičnimi robotskimi podatki. Vendar pa način, na katerega se običajno uporablja osnovni RAG, torej pridobivanje posameznega najbolj podobnega primera, ni optimalen [20]. Kakovost pridobljenega primera je odvisna od natančnosti vgraditvenih (embedding) modelov, ki niso popolnoma natančni, saj so bistveno manjši od tipičnih VLM-jev. Da bi izboljšali tako rezultate strategij pozivanja kot tudi napovedovanje dejanj, predlagamo VLM-voden RAG sistem, ki vključuje naslednje glavne komponente:

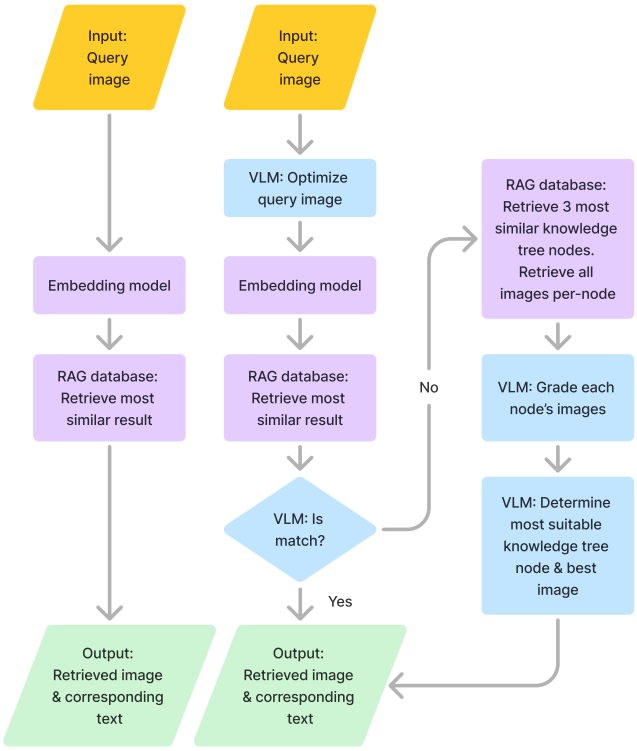
1. **Optimizacija RAG slikovne poizvedbe**
   VLM izboljša relevantnost poizvedbene slike tako, da iterativno izrezuje nerelevantne dele slike, kar zagotavlja, da vgraditveni model prejme osredotočen vizualni vhod in izboljša svojo učinkovitost pri priklicu. Ta korak se uporabi tudi za izboljšanje slik, shranjenih v RAG podatkovni bazi.

2. **Priklic na osnovi podobnosti** Obrezana poizvedbena slika se vgradi z modelom za slikovno vgradnjo, sistem pa iz vektorske podatkovne baze in strukturiranega drevesa znanja pridobi najbolj podobne slikovno-besedilne podatkovne točke.

3. **VLM-voden RAG**
   VLM se uporabi za iterativno ponovno rangiranje pridobljenih rezultatov s primerjanjem poizvedbene slike s kandidati (slikami in besedilom) ter po potrebi za razširitev iskanja. S tem dosežemo znatno povečanje končne natančnosti priklica v primerjavi s standardnim RAG.

Takšna zasnova nam omogoča izkoriščanje visokostopenjskega vizualnega in besedilnega sklepanja VLM-ov za dopolnitev manjših modelov za slikovno vgradnjo. Primerjava našega pristopa z osnovnim RAG je prikazana na Sliki 2.

Za vrednotenje natančnosti napovedovanja dejanj pri VLM-ih v zero-shot načinu ter z uporabo predlaganih pristopov, temelječih na RAG, smo zbrali podatkovno zbirko za napovedovanje dejanj, ki izhaja iz robotskega razstavljanja odpadne elektronike, in zajema različne naloge ter možnosti dejanj.



Slika 2: Primerjava osnovnega RAG s predlaganim VLM-vodenim RAG pristopom. Dodatni elementi so prikazani v modri barvi.

2.2 Primer napovedovanja dejanj pri razstavljanju elektronskih naprav

Nabor podatkov obsega 196 slik, ki predstavljajo stopnje razstavljanja šestih naprav za elektronsko odpadke: štiri vrste detektorjev dima in dve vrsti kalorimetrov. Vsaki sliki so priloženi metapodatki, vključno z imenom naprave, vrsto, prejšnjim dejanjem razstavljanja, naslednjim dejanjem, seznamom preostalih korakov in zastavico za dokončanje.

Na podlagi nabora podatkov smo definirali naslednje štiri naloge za ocenjevanje sposobnosti modelov za interpretacijo proceduralnih vizualnih podatkov:

- Predvidevanje prejšnjega koraka: Ugotovi prejšnji korak, ki je pripeljal do trenutnega stanja na sliki.
- Predvidevanje naslednjega koraka: Predvidi naslednje dejanje v zaporedju razstavljanja.
- Predvidevanje preostalih korakov: Predvidi korake, potrebne za dokončanje procesa razstavljanja.
- Predvidevanje zadnjega koraka: Ugotovi, ali trenutno stanje predstavlja končni korak razstavljanja.

2.3 Rezultati in diskusija

Razvili smo pristop za napovedovanje robotskih dejanj z uporabo VLM skupaj z lokalno bazo podatkov in RAG, kjer VLM najprej izboljša vhodno sliko poizvedbe RAG, nato iterativno poizveduje

po bazi podatkov in določi najprimernejše ustrezne slike in besedilno vsebino. Pridobljeno najboljšo sliko in povzetek besedila je VLM uporabil v kontekstu napovedovanja naslednjega dejanja v robotski delovni celici. Naš pristop RAG, voden z VLM, smo primerjali z osnovnim pristopom, ki ne uporablja RAG (nič posnetkov), pristopom, ki uporablja eno samo najboljšo pridobljeno sliko RAG (Single-RAG), in pristopom, kjer je modelu namerno dodeljena zavajajoča slika in ustrezno besedilo (Bad-Single-RAG). Podrobni rezultati so na voljo v pripetem članku, spodaj pa so izpostavljeni ključni povzetki.

Prvič, korak izboljšanja poizvedbe slike se je izkazal za uspešnega pri izboljšanju baze podatkov RAG in slike poizvedbe z odstranitvijo nepotrebnih elementov, hkrati pa ohranja ustrezne elektronske naprave znotraj izrezane slike. Ne glede na uporabljeni model vdelave slike RAG to vodi do povečane učinkovitosti iskanja.

Drugič, ovrednoteni so bili različni modeli vdelave slik z odprto kodo. Pokazalo se je, da korak izboljšanja poizvedbe po slikah znatno izboljša učinkovitost iskanja osnovnega sistema RAG. Poleg tega večji modeli vgrajevanja kažejo višjo natančnost iskanja top-1 in top-n.

Tretjič, ocenjena je bila učinkovitost napovedovanja dejanj z ničelnim številom poskusov tako odprtokodnih kot komercialnih modelov. Primerjana je bila s pristopom Bad-Single-RAG, kjer so bili modeli namerno podarjeni zavajajoči sliki, da bi simulirali primer, ko baza podatkov RAG ne vsebuje ustreznih podatkovnih točk. Pokazalo se je, da se v tem primeru učinkovitost modelov ne poslabša.

Izkazalo se je, da pristop Single-RAG znatno poveča natančnost napovedovanja dejanj v primerjavi z napovedmi z ničelnim številom poskusov. Ključno je, da je privedel do minimalnega povečanja zakasnitve napovedi. Natančno nastavljanje ni bilo učinkovito na našem naboru podatkov zaradi njegove majhnosti in so ga prekašali pristopi, ki temeljijo na RAG.

Nazadnje je bil ocenjen pristop RAG, voden z VLM, ki je pokazal povečanje tako natančnosti iskanja kot tudi natančnosti napovedovanja končnega dejanja. Ta pristop je bil primerjan z metodo Single-RAG in je pokazal statistično signifikantno povečanje natančnosti napovedovanja dejanj za vse modele na račun podaljšanja latence napovedovanja, saj čas, potreben za izvedbo koraka RAG, vodenega z VLM, ni deterministične.

Pristopa optimizacije slikovnih poizvedb in RAG, voden z VLM, znatno povečata operativno avtonomijo robotske delovne celice, kar omogoča njeno hitro prilagoditev razstavljanju novih tipov naprav brez nenehnega natančnega uglaševanja modela, kar se je izkazalo za neučinkovito pri manjših naborih podatkov, kot v našem primeru.

3 Analiza strategij pozivanja

Cilj te analize je oceniti učinkovitost različnih strategij pozivanja (an. *prompting strategy*) za omogočanje pozivanja z neposrednim (zero-shot) učenjem. Primerjamo strategiji pozivanja za eno nalogo (an. *single task*, ST), in več nalog (an. *multi task*, MT), da ugotovimo kakšen vpliv imata na natančnost modelov.

3.1 Metodologija

Slikovno-jezikovne modele smo ovrednotili na štirih nalogah razstavljanja naprav (definiranih zgoraj), uporabili pa smo dve različni strategiji pozivanja. Vsaka strategija je bila implementirana v

nastavitvi pozivanja z neposrednim napovedovanjem, t.j., modeli niso bili fino prilagojeni našemu specifičnemu naboru podatkov.

3.2 Strategije pozivanja

Implementirali smo dve strategiji pozivanja:

- **Pozivanje za eno nalogo (ST)**: Za vsako od štirih nalog so izdani samostojni pozivi. Vsak poziv vključuje ime naprave, urejen seznam možnih korakov razstavljanja za to napravo in sliko trenutne stopnje razstavljanja. Model vrne po eno napoved za vsako nalogo v datoteki JSON.
- **Pozivanje za več nalog (MT)**: Zgradili smo enoten poziv za zahtevo napovedi za vse štiri naloge hkrati. Ta poziv vključuje ime naprave, urejen seznam možnih korakov razstavljanja in sliko trenutne stopnje razstavljanja. Model eno datoteko JSON, ki vsebuje vse štiri napovedi. S tem pristopom želimo izkoristiti potencialne odvisnosti med nalogami v procesu razstavljanja.

3.3 Eksperimentalna postavitev in ocenjevalne metrike

Vrednotenje smo izvedli na podlagi posameznih slik. Vsaka od 196 slik je bila uporabljena za generiranje napovedi za vse štiri naloge z uporabo obeh strategij pozivanja. Uporabili smo sledeče štiri 4-bitno kvantizirane (Q4) VLM: LLaVA 34B, Llama-3.2-Vision 90B, Gemma-3 27B in Qwen-2.5 72B.

Vsi modeli so bili nameščeni na lokalnem strežniku, opremljenem s štirimi grafičnimi karticami NVIDIA RTX 4090 24GB. Uporaba virov se je razlikovala glede na model: Gemma-3 je zahtevala eno grafično kartico, LLaVA je zasedla dve grafični kartici, medtem ko sta Llama in Qwen uporabljala vse štiri grafične kartice. Vsak model je bil testiran desetkrat z enakimi konfiguracijami, da se zagotovi stabilnost in upošteva potencialna varianca.

Izhod modelov smo razčlenili iz datotek JSON in izračunali natančnost za vsako nalogo. Za primerjavo je bil ocenjen tudi izhodiščni (t.i., “dummy”) klasifikator, ki je naključno izbral napovedi iz nabora možnih korakov razstavljanja.

3.4 Rezultati in diskusija

Tabela 1 prikazuje rezultate vrednotenja, ki kažejo, da pozivanje za več nalog dosledno prekaša pozivanje za eno nalogo pri zero-shot razstavljanju naprav. S pozivanjem za več nalog v povprečju natančnost izboljšamo za 17,95 odstotnih točk. Izboljšavo pripisujemo sposobnosti modela, da izkoristi odvisnosti med nalogami in bolj dosledno razmišlja o procesu razstavljanja.

Opazimo, da natančnost modela ni bila neposredno povezana s številom parametrov. Najmanjši model, Gemma3-27B, je dosegel najvišjo splošno natančnost (65,42 %), ko je uporabljal pozivanje za več nalog in skoraj podvojil natančnost v primerjavi s pozivanjem za eno nalogo. Po drugi strani pa je največji model, Llama3.2-Vision-90B, deloval slabše od izhodiščnega klasifikatorja.

Primerjava posameznih nalog razkrije jasno hierarhijo težavnosti. “Zadnji korak” je najpreprostejša nalogo, kjer sta Gemma3-27B in Qwen2.5-72B dosegla skoraj popolno natančnost

	Prejšnji korak		Naslednji korak		Preostali koraki		Zadnji korak		Povprečje	
Model	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**
Gemma-3 27B	17.39 (0.00)	**54.64 (1.21)**	25.47 (1.52)	**53.86 (1.24)**	18.63 (1.97)	**53.86 (1.24)**	73.91 (0.00)	99.32 (0.25)	**33.85**	**65.42**
LLaVA 34B	23.48 (4.43)	31.97 (2.75)	5.65 (5.85)	32.51 (2.46)	21.74 (3.37)	17.64 (2.22)	63.48 (7.58)	76.80 (2.32)	28.59	39.73
Llama-3.2-Vision 90B	24.78 (5.52)	23.98 (3.12)	7.39 (3.91)	18.15 (3.14)	24.35 (3.48)	18.29 (3.95)	62.61 (9.16)	52.61 (5.18)	29.78	28.26
Qwen-2.5 72B	24.78 (4.78)	43.77 (1.66)	19.56 (5.91)	43.77 (1.66)	26.09 (0.00)	43.77 (1.66)	51.30 (5.07)	**99.92 (0.23)**	30.43	57.81
Izhodiščni klasifikator	20.57 (1.84)		20.51 (1.84)		35.38 (2.97)		50.52 (2.94)		31.75	

Tabela 1: Primerjava natančnosti pri strategijah pozvianja z eno nalogo (ST) in več nalogami (MT), povprečje čez 10 ponovitev za 4 naloge razstanljanja (v oklepaju standardna deviacija).

(>99 %) pri uporabi pozivanja za več nalog. Nasprotno so “Preostali koraki” najtežja naloga, kjer sta modela LLaVA in Llama pokazala slabše delovanje kot izhodiščnegi klasifikator. To kaže, da se nekatere modelne arhitekture morda težko spopadajo s kompleksnostjo te naloge, zlasti v kombinaciji s pozivanjem za več nalog.

Vrednotenje prikazuje, da pozivanje za več nalog omogoča VLM-jem, da učinkovito izkoristijo odvisnosti med nalogami, kar vodi do izboljšanega zaporednega razmišljanja in višje natančnosti pri robotskih nalogah razstavljanja. Te ugotovitve prikažejo praktične smernice za zasnovo pozivov v kontekstu uporabe VLM pri robotskih nalogah razstavljanja.

4 Zakjučki

Pristopa optimizacije slikovnih poizvedb in RAG, voden z VLM, znatno povečata operativno avtonomijo robotske delovne celice, kar omogoča njeno hitro prilagoditev razstavljanju novih tipov naprav brez nenehnega natančnega uglaševanja modela, kar se je izkazalo za neučinkovito pri manjših naborih podatkov, kot v našem primeru.

Metodologija je splošno uporabna v procesih, kjer se izvaja sekvence nalog. V primeru DIGITOP je to lahko pri momdularnem sestvljanju robotov ali pa pri prilagajanju operacij za vizualno kontrolo kvalitete, kar se izvaja s partnerji projekta.

Literatura

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[2] T. Deleu and Y. Bengio. The effects of negative adaptation in model-agnostic meta-learning. *arXiv preprint 1812.02159*, 2018.

[3] T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T. R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Žiga Gosar, I. Kovač, and A. Ude. Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. *Robotics and Computer-Integrated Manufacturing*, 66:101979, 2020.

[4] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld. Pddl - the planning domain definition language. 08 1998.

[5] N. Jaquier, M. C. Welle, A. Gams, K. Yao, B. Fichera, A. Billard, A. Ude, T. Asfour, and D. Kragic. Transfer learning in robotics: An upcoming breakthrough? a review of promises and challenges. *The International Journal of Robotics Research*, 44(3):465–485, 2025.

[6] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020.

[7] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[8] R. Lakatos, P. Pollner, A. Hajdu, and T. Joó. Investigating the performance of retrieval-augmented generation and domain-specific fine-tuning for the development of ai-driven knowledge-based systems. *Machine Learning and Knowledge Extraction*, 7(1), 2025.

[9] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu. Towards generalist robot policies: What matters in building vision-language-action models, 2024.

[10] Z. Li, X. Wu, H. Du, F. Liu, H. Nghiem, and G. Shi. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges, 2025.

[11] Z. Lončarević, G. Li, G. Neumann, and A. Gams. An encoder-decoder architecture for smooth motion generation. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 358–366. Springer, 2023.

[12] M. Mavsar, M. Denisa, B. Nemec, and A. Ude. Intention recognition with recurrent neural networks for dynamic human-robot collaboration. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 208–215, Dec 2021.

[13] M. Mortaheb, M. A. A. Khojastepour, S. T. Chakradhar, and S. Ulukus. Rag-check: Evaluating multimodal retrieval augmented generation performance, 2025.

[14] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters. Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9, 2022.

[15] P. Nimac and A. Gams. Determining sample quantity for robot vision-to-motion cloth flattening. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 3–11. Springer, 2024.

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[17] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

[18] S. Towhidul, S. M. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das. A comprehensive survey of hallucination mitigation techniques in large language models, 2024.

[19] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui. Retrieval-augmented generation for ai-generated content: A survey, 2024.

[20] X. Zheng, Z. Weng, Y. Lyu, L. Jiang, H. Xue, B. Ren, D. Paudel, N. Sebe, L. V. Gool, and X. Hu. Retrieval augmented generation and understanding in vision: A survey and new outlook, 2025.

Dodatek

- Improving Retrieval Augmented Generation for Robot Action Prediction using VLMs, avtorjev Boris Kuster, Nikola Marić, Fatima Aziz, Boshko Koloski, Andrej Gams, Aleš Ude; v pripravi.
- Zero-Shot Benchmarking of Vision-Language Models for Device Disassembly, avtorjev Fatima Aziz, Nikola Marić, Boris Kuster, Aleš Ude, and Boshko Koloski; sprejeto v objavo na konferenci Automation, Robotics & Communications for Industry 4.0/5.0/6.0 (ARCI' 2026).

Improving Retrieval Augmented Generation for Robot Action Prediction using VLMs

Boris Kuster, Nikola Marić, Fatima Aziz, Boshko Koloski, Andrej Gams, Aleš Ude

***Abstract*— Action prediction is needed in many robotic applications to enable the operation of robots in uncertain and dynamic environments. The recently developed large vision-language models (VLMs) provide a promising methodology, since they are based on large datasets of general knowledge. The key limitation when applying VLMs in robotic applications is that the datasets are usually too small to allow for direct fine-tuning of existing VLMs. In this paper, we present an approach to action prediction with VLMs based on Retrieval-Augmented Generation (RAG), a technique which increases prediction accuracy in VLMs by dynamically appending relevant examples from a local database to the model's input. We improve upon the baseline RAG approach (retrieval of the single most similar example) by using the VLM to incrementally improve the RAG database, optimize the region-of-interest of a query image, expand the RAG search where necessary, and re-evaluate the retrieved elements, thus improving retrieval performance. The approach is evaluated on robot-action prediction in electronic waste recycling and demonstrates a statistically significant absolute accuracy improvement of 45% over zero-shot predictions and 20% over the baseline RAG approach.**

I. Introduction

Standard methodologies for electronic waste recycling rely on a crush-and-separate approach, where devices are first crushed, followed by a physio-chemical separation of materials into useful categories. However, this approach may cause fires if dangerous components such as lithium-ion batteries are crushed [1]. Such components should therefore be removed before crushing each device. This requires their disassembly. Here a problem arises that in the recycling of electronic waste, it is necessary to deal with a wide variety of electronic items, be it from the same device family (e.g. smoke detectors) or even across different families of devices (e.g. smoke detectors, heat cost allocators, remote controls, etc.). Consequently, it is not feasible to plan every disassembly operation that may be necessary in advance. The devices may appear similar, however, the disassembly sequence can be entirely different. The correct disassembly sequence may also depend on the state of the device, as the device may be partially damaged. Thus, it is necessary to plan robotic actions on the fly, or in other words, a recycling robot must be able to automatically predict robot disassembly actions based on the current state of the device.

Vision is crucial to implement automated disassembly of electronic devices. However, the available datasets are often small in size, increasing the difficulty of training object detection models [2]. In recycling, it is constantly necessary to deal with new devices that have not been used for training vision algorithms. Thus, it is necessary that the trained models are robust with respect to new devices and that they can be updated with new devices if necessary, with little or no additional training.

Recently, large Vision-Language Models (VLMs) have received increased attention in the field of robotics due to their substantial embedded world knowledge, which gives the possibility of considering general knowledge that is hard to incorporate or not available in classic planners and planning domains, e.g., in Planning Definition Domain Language (PDDL) [3]. VLMs also have the ability to process one or more images and reason about objects and their relations [4]. A key advantage of VLMs (and LLMs) is the possibility of appending relevant examples to the input prompt, which is known as few-shot or in-context learning [5]. While these relevant examples can be specified per-task by a programmer, a more general approach entails retrieving them dynamically from a database, based on the similarity of current inputs to the database elements. This approach is known as Retrieval-Augmented Generation (RAG) [6] and allows VLM performance to be improved without computationally intensive fine-tuning, while also avoiding the possibility of catastrophic forgetting and hallucination [7].

The idea of RAG is to have a local database of data points, where each data point can contain an image, text, and other modalities, or a combination thereof, as shown in Fig. 1.

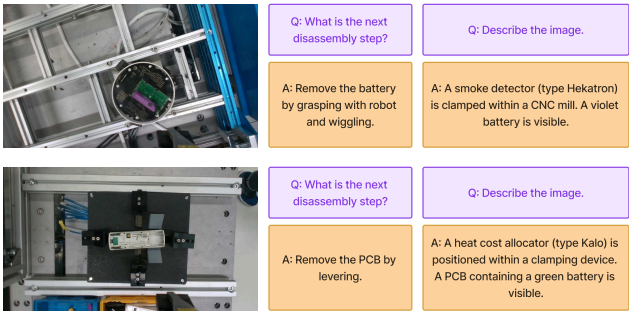


Fig. 1. Example of two RAG data points containing an image and text - a set of question-answer pairs, used in disassembly action prediction.

These data points are then embedded into the vector space by means of a modality-specific embedding model, typically a much smaller model than a VLM [8]. When a downstream model, e.g. a VLM, is requested to perform inference on a novel data point, elements of the input data point, e.g. the input image, are embedded by the embedding model, and relevant examples (image & corresponding text) are retrieved

All authors are with Department for Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia
{boris.kuster}@ijs.si

from the database and added to the VLM prompt. This approach increases the prediction accuracy of VLMs [9]. Especially for smaller datasets, RAG can be more effective than fine-tuning [10]. Some important advantages of using a RAG system are that adding data points is not programming-intensive, the performance of the downstream model can be increased, and the database can also be used as a fine-tuning dataset if necessary.

In a baseline RAG approach, only the most similar data point is retrieved from the database and added to the VLM prompt. This has several potential downsides:

1) The database may lack relevant examples, and including irrelevant data points may decrease prediction accuracy.
2) The most similar data point in the embedding vector space may not be the most representative, since embedding models are not perfectly accurate.
3) The input query may contain distractor elements, e.g., an image where the relevant object only takes up a small portion of the input image, decreasing the embedding quality.
4) In case of text-based retrieval, embedding models may struggle on unstructured or longer text segments.

To mitigate these issues, we present an algorithm - VLM-guided RAG, where a VLM is used to improve (crop) the input query image and the images within the RAG database, re-evaluate the retrieved results, and optionally extend the RAG search until the VLM determines a suitable data point is found, or a pre-set time limit has been exceeded. This process generates an informative example that can be used by downstream models. In the context of robotics, especially when applied to dynamic applications such as recycling, this can improve the performance of VLMs for next-action prediction and similar tasks. We demonstrate its effectiveness for action prediction in the task of disassembly of waste electronics. Such AI-based approaches can significantly reduce the programming workload required to adapt robotic workcells to new applications [11]. To evaluate the effectiveness of our approach, we compare the VLM-guided RAG approach with several baselines: baseline single-retrieved element RAG and a non-RAG (zero-shot) approach, where the VLMs are not given any additional helpful data. Each method is evaluated using both open-source and commercial VLMs, allowing us to assess performance across different model types and retrieval strategies.

II. Methodology

Action prediction in robotics is a key challenge in enabling greater automation of robotic workcells and decreasing the programming workload when applying them to new tasks, ultimately supporting robots in performing a more diverse and extensive set of tasks. Traditional approaches have relied on neural network models such as Multi-Layer Perceptrons (MLPs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory networks (LSTMs) [12]. Although effective for specific cases, such models often suffer from limited generalization and require task-specific training.

Recently, VLMs have shown impressive results in this field, presenting a novel approach to next action prediction in robotic workcells due to their generalization capability, large pre-training datasets, and ability to interpret visual inputs [13]. However, VLMs are rarely trained on similar data as seen in robotic action prediction. Fine-tuning them remains challenging due to the limited size of available action prediction datasets and the substantial computational cost required to train larger models, which generally outperform their smaller counterparts [14].

To improve the accuracy of action prediction without fine-tuning, one practical solution is to use a local database of examples via RAG. This approach enables the model to retrieve and reference relevant examples during inference, bridging the gap between general-purpose VLM knowledge and domain-specific robotic data.

However, the way in which baseline RAG is commonly utilized, i.e., to retrieve the single most similar example, is not optimal [15]. The quality of the retrieved example depends on the accuracy of the retrieval (embedding) models, which are not perfectly accurate, since they are much smaller than typical VLMs. To improve both the retrieval and action prediction performance, we propose a VLM-guided RAG system, which consists of the following main components:

1) RAG Image-Query optimization (Sec. II-B)
   A VLM improves the relevance of the query image by iteratively cropping out irrelevant image regions, ensuring that the embedding model receives a focused visual input and improving its retrieval performance. This step is also used to improve the images stored in the RAG database.
2) Similarity-Based retrieval (Sec. II-C)
   The cropped query image is encoded using an image embedding, and the system retrieves most similar image-text data points from a vector database and a structured knowledge tree.
3) VLM-Guided RAG (Sec. II-D)
   A VLM is used for iteratively re-ranking the retrieved results by evaluating the query image and the retrieved candidate image and text, and expanding the search where necessary, resulting in a significant increase in final retrieval accuracy compared to standard RAG.

This design allows us to leverage the high-level visual and text reasoning capabilities of VLMs to augment the smaller image embedding models. A comparison of our approach to baseline RAG is shown in Fig. 2.

For evaluation of the action prediction performance of VLMs in a zero-shot manner, as well as with the proposed RAG-based approaches, we collected an action prediction dataset originating in robotic disassembly of waste electronics, described in Sec. III-A, encompassing a variety of tasks and action possibilities.

A. VLM tool calling

To ensure that the VLM gives us exactly the type of answer we need for the particular pipeline steps (e.g., Image Query Optimization, VLM-guided RAG) – in the exact structured

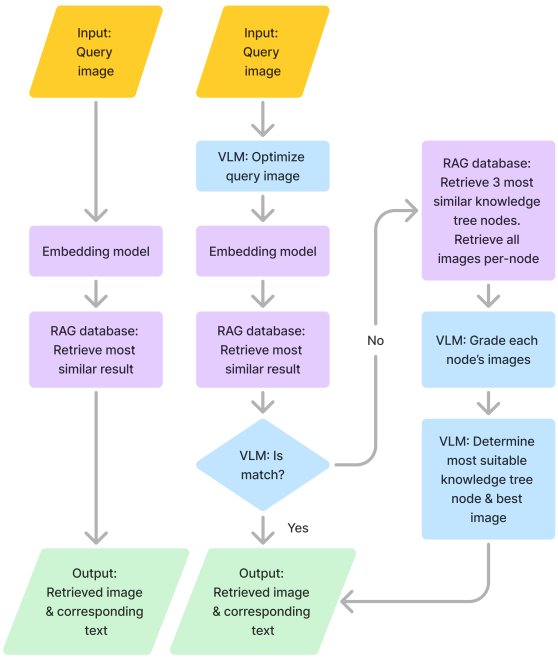


Fig. 2. Comparison of baseline RAG with the proposed VLM-guided RAG approach. The additional elements are shown in blue.

format we expect — we use a method known as tool calling [16]. In tool calling, the programmer (us) defines a set of "tools" the VLM is allowed to use. Each tool defines:

1) a name (e.g., *set_grid_size*),
2) parameters it accepts (e.g., *grid_size*),
3) the type of each parameter (e.g., integer, string).

These definitions are given to the VLM before it generates a response. Guided decoding [17] is then used to ensure that the model's output exactly follows this predefined schema — so we always get a valid, structured, machine-readable response. The approach also allows for real-time adaptability, since no model fine-tuning is required in the event that tool definitions change or new tool options are added. We apply this process in our RAG Image-Query Optimization and VLM-guided RAG pipeline, as well as in next-action prediction, to ensure that every VLM output is not only correct in content but also appropriately structured for the next step in the workflow.

B. RAG Image-query optimization

The performance of a RAG system depends strongly on the quality of its input query (in our case, an image). Existing work has focused on optimizing text-based RAG [18] and on optimizing vision-based RAG for document retrieval [19], whereas query optimization for vision-based RAG in robotics remains largely unexplored.

To improve the quality and retrieval performance of the RAG database, we design an input image optimization (cropping) pipeline, which uses a VLM to improve the relevance of the query image. Since the query image may contain irrelevant elements, a VLM is used to crop the image, i.e., determine a region of interest (ROI) recursively. The cropped image is passed to the image encoder, improving the quality of the vector embedding, which increases the relevance of the retrieved images. This is also used for incremental RAG database improvement, as it is crucial that both database and query images contain maximally relevant visual information. Without consistent cropping on both sides, semantic mismatches may occur—for example, a database image may show an entire table with a small smoke detector in the background, while the cropped query image shows only the detector itself. This discrepancy leads to suboptimal encoder performance due to mismatched visual focus and embedding space alignment.

Despite some recent VLMs (e.g., Qwen3-VL [20]) supporting object detection and being able to output bounding boxes, the object localization accuracy is still low, particularly in the case of novel objects not present in the dataset. However, VLMs have been shown to be able to make effective use of image annotations, such as drawn-on circles, grids, and numbers. One such approach is Scaffolding [21], whereby a dot matrix is overlayed on the image as visual information anchors along with multidimensional coordinates as textual positional references.

We designed and evaluated two iterative cropping approaches - grid-based and bounding-box-based- that enable the VLM to focus on the relevant parts of the image. At each iteration, the VLM receives (i) the full-size query image, (ii) the temporary crop version of the query image, and (iii) a text instruction (e.g., "Crop the image so the electronic device is centered. Keep relevant neighboring information, such as the device within which the electronic device may be mounted."). This RAG optimization prompt instructs the VLM to crop the image to the region described in the text while keeping relevant neighborhood information, such as if the object is clamped within a machine. The VLM is also given a list of tools to call, along with their parameters, to ensure parsable output. The resulting cropped images are then passed to the image embedding model during RAG similarity-based search.

1) Grid-based cropping: Grid-based image cropping is implemented by augmenting the input query image with a grid, along with a single integer to denote the ID of each grid rectangle.

We define the following tools which the VLM can use:

- crop: crop by selecting top left and bottom right grid element indices, extract a rectangular portion of the image,
- set grid size: change the grid's number of rows and columns,
- reset: Reset the temporary cropped image to the original image,
- finish: finish the task and return the final cropped image, along with the bounding box.

In an iterative procedure, shown in Algorithm 1, the VLM selects a tool and arguments until the final crop of the image is achieved, as determined by the VLM. If the grid lines occlude the relevant image parts, the VLM can change the

number of rows and columns of the grid. The VLM can crop the image by selecting the IDs of the top-left and bottom-right grid elements. Since the original input query image is always included in the prompt, the VLM can choose to reset the temporary cropped image to the original image in the case of a bad crop. Lastly, the VLM is instructed to select the "finish" tool when the image is sufficiently cropped, and we impose a fixed maximum number of iterations to guarantee termination. An example of an input image, grid-annotated image, and the final crop are shown in Fig. 3.

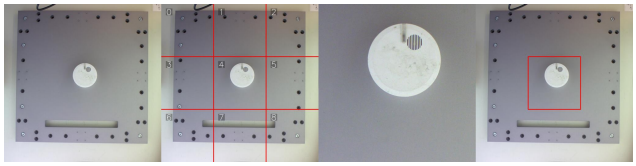


Fig. 3. The input query image, grid-labeled image, final cropping result, and the cropped area shown in the original image.

During the iterative cropping steps, the image is downscaled to a predefined smaller size to enable faster VLM computation. However, the original high-resolution image is retained, and the final crop is taken from this original image, ensuring that the resulting cropped image does not lose detail.

Algorithm 1 Grid-based optimization of image query with a VLM

procedure GRIDCROP(q, o) ▷ Query image, text objective
  $c \leftarrow copy(q)$ ▷ Temporary crop
  $g \leftarrow (3, 3)$ ▷ Grid size
  $t \leftarrow None$ ▷ Tool
  $p \leftarrow None$ ▷ Tool parameters
  $b \leftarrow None$ ▷ Bounding box
  $i \leftarrow 0$ ▷ Current iteration
  while ($t \neq Finish$) and ($i \leq 5$) **do**
    $c \leftarrow overlayGrid(c, g)$
    $t, p \leftarrow vlm(q, c, o)$
      ▷ VLM selects tool and parameters
    if t is $crop$ **then**
      $b, c \leftarrow crop(c, p, g)$
        ▷ Crop based on selected grid indices
    if t is $setGridSize$ **then**
      $g \leftarrow p$
    if t is $reset$ **then**
      $c \leftarrow q$
    $i \leftarrow i + 1$
  return c, b
    ▷ Return cropped image and bounding box

2) Bounding-box-based cropping: The second approach evaluated is iterative bounding-box prediction. Here, the VLM outputs normalized top-left bounding-box coordinates, as well as normalized bounding-box width and height. Compared to the grid-based approach, this offers more fine-grained and flexible localization, which we hypothesize will lead to improved performance. The VLM is provided with the following tools:

- set bounding box parameters: specify bounding box based on input normalized top left (x,y) coordinates, bounding box width, and bounding box height,
- finish: finish task and return final bounding-box cropped image.

The iterative process enables the VLM to further refine the bounding box based on analyzing the image and the currently selected bounding box. Similarly to the grid-based approach, a maximum number of iterations is imposed to guarantee termination. The process is shown in Algorithm 2.

Algorithm 2 Bounding-box-based optimization of image query with a VLM

procedure BBOXCROP(q, o) ▷ Query image, text objective
  $c \leftarrow copy(q)$ ▷ Temporary crop
  $t \leftarrow None$ ▷ Tool
  $p \leftarrow None$ ▷ Tool parameters
  $b \leftarrow None$ ▷ Bounding box
  $b \leftarrow None$ ▷ Bounding box
  $i \leftarrow 0$ ▷ Current iteration
  while ($t \neq finish$) and ($i \leq 5$) **do**
    $t, p \leftarrow vlm(q, c, o)$
      ▷ VLM selects tool and parameters
    if t is $setBoundingBoxParameters$ **then**
      $b, c \leftarrow drawBoundingBox(c, p)$
    $i \leftarrow i + 1$
  return c, b
    ▷ Return cropped image and bounding box

C. RAG database & lookup

Our RAG database combines two key components: a structured knowledge tree and a vector database. These enable efficient and accurate multimodal retrieval for downstream tasks, e.g. disassembly action prediction. The vector database facilitates similarity-based image retrieval, allowing the system to find visually relevant data points based on an input query image. Meanwhile, the knowledge tree provides a hierarchical organization of data points (devices and their disassembly sequences), linking each image to structured textual information and also enabling an overview of device types present in the database.

1) Vector database and similarity-based image retrieval: The similarity-based RAG approach retrieves relevant database data points (which can contain text, images, or other modalities) based on an input query (e.g. image, text, ...). To perform similarity-based RAG lookup, an embedding model is used to determine similarity between a query data point and database data points.

In our case, a single data point contains both text and an image. Therefore, as we perform image-based search, corresponding text is also retrieved, which assists the VLM

in making accurate next-action predictions. This text includes the device name and type, which serve as entry points for follow-up lookups in the structured knowledge tree, allowing the system to retrieve the full disassembly sequence and related device information.

In general, modality-specific embedding models can be used to retrieve similar elements of any modality (e.g. text, images, videos, sound). For our experiments, we perform image-based search. To retrieve candidate images (and corresponding text) from a database based on an input query image, an image embedding model *emb* is utilized. Some commonly used image embedding models include *CLIP* [8], *ResNet* [22], and *VisionTransformer* [23]. While *ResNet* and *VisionTransformer* are primarily used as image classification models, they can still produce embedding vectors for images.

The embedding model *emb* with parameters θ calculates an embedding vector $\mathbf{x}$ for the query image q, which is then compared to vectors $\mathbf{y}$ generated from database images d using cosine similarity s, as shown in Eq. (1):

$$\begin{aligned} \mathbf{x} &= \mathrm{emb}_{\boldsymbol{\theta}}(\mathrm{q}) \\ \mathbf{y} &= \mathrm{emb}_{\boldsymbol{\theta}}(\mathrm{d}) \\ s &= \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \end{aligned} \tag{1}$$

Top n most similar database elements are returned, where the parameter n can be set arbitrarily. An example of a query image and the corresponding retrieved images is shown in Fig. 4.

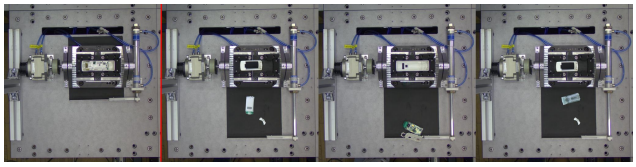


Fig. 4. Example of RAG-retrieved results, where the left-most image is the query.

2) Knowledge tree: A downside of a standard vector database is that data points cannot be hierarchically structured. For robotic disassembly action prediction, however, it is useful to organize devices into families and to keep all disassembly steps for a specific device grouped together. To this end, we utilize a knowledge tree which provides a hierarchical organization of devices and their disassembly sequences. Each knowledge tree's leaf node represents a particular device (e.g., smoke detector Kalo), while the parent node represents the device family (e.g., smoke detector). For each disassembly step, an image along with the corresponding ground truth text is stored within the device leaf node. The corresponding text contains all relevant information for action prediction, such as information about the device name and type, next required disassembly action, and the remaining sequence of disassembly steps. The corresponding images are embedded in the vector database and connected to the knowledge tree. Therefore, when an image is returned using a vector search, the leaf name is also included in the retrieved text. This allows the VLM to retrieve all disassembly steps for a particular device if the *leaf name search* tool is used. Fig. 5 shows an example of the knowledge tree structure.

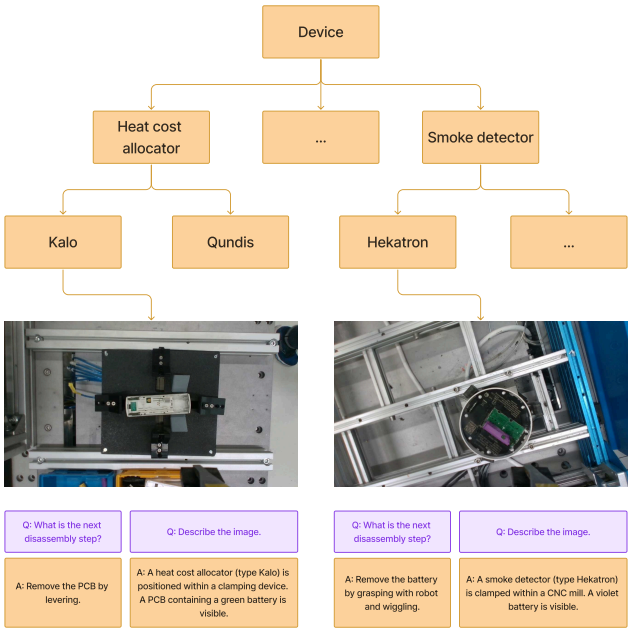


Fig. 5. Database structure showing the knowledge tree along with relevant images & text.

D. VLM-guided Retrieval Algorithm

In many cases, particularly for robotic action prediction, retrieving only the top-1 most similar data point may be insufficient due to imperfect embedding model performance. While the RAG database allows for retrieving an arbitrary number of data points (i.e., images & corresponding text), simply appending several data points to the VLM prompt is ineffective for the following reasons:

1) the prediction latency is significantly increased when processing multiple high-resolution images, and
2) model performance is shown to degrade with longer prompt lengths, a problem known as "context stuffing" [24]. In case of multiple images, the VLM may lose focus on the query image.

For these reasons, it is better to separately conduct a more detailed search step and include only the most representative example data point. To further increase RAG retrieval accuracy, and by extent, robot prediction accuracy, while also using the information embedded in the knowledge tree, we develop a novel retrieval algorithm, VLM-guided RAG. Since processing multiple images using VLMs is slow, we aim to preserve the computational efficiency of RAG to decrease prediction latency. Therefore, the first step in our algorithm is re-evaluating the most-similar RAG-retrieved candidate image in relation to the query image. The VLM is instructed to provide a positive or negative score to the

image, based on location similarity (e.g. CNC machine, flat table, ...), object shape similarity, and internal electronic component similarity, as well as the ground-truth previous disassembly steps of both candidate and query images.

If the VLM confirms the image, the database image and corresponding text are directly passed to the action prediction VLM, ensuring minimal additional processing time.

In the case of the VLM rejecting the image, we retrieve $n = 15$ most-similar images from the RAG database. Then, we determine which leaf nodes (i.e., devices) the images belong to and deduplicate the leaves, keeping the $m = 3$ most relevant knowledge tree leaves. Afterwards, all the images corresponding to each leaf are retrieved along with their corresponding text.

For each retrieved leaf and its images, the VLM grades each of them sequentially, before providing a final score (0-10) of the leaf similarity and a summary of findings. For each leaf, the most-relevant image is stored along with the corresponding text. Optionally, the VLM can extend the search to a maximum of $n = 5$ leaves. The process is shown in Algorithm 3.

The VLM is provided with the following tools:

- Boolean grade image: Provide a positive or negative score for an image, along with reasoning. Used for grading the top-1 RAG-retrieved image.
- Expand search by one leaf: Retrieve all images & text of the following most-similar knowledge tree leaf.
- Grade image: Provide a 0-10 score of a candidate image, along with reasoning.
- Grade leaf: Provide a 0-10 score of a leaf node, based on textual reasonings about a particular leaf's images.

In case of highest score collisions (i.e., two images from same or different nodes being given the same highest score), a three-image grading step is performed, based on the query image and the 2 candidate images, where the VLM selects the best candidate image.

The final output of our VLM-guided RAG algorithm is the single most relevant retrieved image and corresponding text. Additionally, reasoning for the steps of the grading process is provided to improve VLM performance using the Chain-of-thought methodology [25]. The retrieved image & text is then passed to the VLM in downstream tasks, such as next-action prediction in dynamic robotic tasks. This search process increases retrieval accuracy over baseline RAG. A comparison of baseline RAG and our VLM-guided RAG approach is shown in Fig. 2.

Algorithm 3 VLM-guided RAG

procedure VLM-GUIDED RAG(q) ▷ Query image q
  $bestImg, bestText, bestScore \leftarrow None$
  $imageSummaries, nodeSummaries \leftarrow [\,]$
  $candidates \leftarrow$ RETRIEVETOPN($q, n = 1$)
  $i, t \leftarrow candidates[0]$ ▷ Top candidate image and text
  $score \leftarrow$ VLM.BOOLSCORE(q, i, t)
    ▷ Re-evaluate top-1 RAG candidate using VLM
  if $score =$ POSITIVE **then**
    return (i, t, NONE, NONE)
      ▷ Directly pass confirmed image and text
  else
    ▷ Retrieve additional similar images for leaf-based expansion
    $E \leftarrow$ RETRIEVETOPN($q, n = 15$)
    $nodes \leftarrow$ DEDUPLICATE(GETNODES(E))
    $m \leftarrow 3$ ▷ Number of searched leaf nodes
    $topNodes \leftarrow$ GETTOPNODES($nodes, m$)
    ▷ Sequential grading of images and summarization per node
    for all $node \in topNodes$ **do**
      $nodeImgs \leftarrow$ GETNODEIMAGES($node$)
      $nodeTexts \leftarrow$ GETNODETEXTS($node$)
      $nodeSummary \leftarrow [\,]$
      for all $(img, txt) \in (nodeImgs, nodeTexts)$ **do**
        $s, summary \leftarrow$ VLM.GRADEIMAGE(q, img, txt)
        $imageSummaries$.APPEND($summary$)
      $nodeScore, nodeSummary \leftarrow VLM.GradeLeaf(imageSummaries)$
      ▷ Select the highest-scoring candidates
    $bestCandidates \leftarrow GetTopScores(imageSummaries)$
    if |BESTCANDIDATES| > 1 **then**
      ▷ Resolve tie with 3-image VLM comparison
      $bestImg \leftarrow VLM.Compare(q, bestCandidates$
+ $nodeSummaries)$
    else
      $bestImg \leftarrow$ BESTCANDIDATES$[0].img$
    ▷ VLM determines whether to expand search further
    if $VLM.ExpandSearch(bestImg, q)$ **then**
      $m \leftarrow m + 1$
      $topNodes \leftarrow GetNode(nodes, m)$
      goto Line 14
    $bestText \leftarrow$ GETCORRESPONDINGTEXT($bestImg$)
    $bestScore \leftarrow$ GETSCORE($bestImg$)
    ▷ Return best retrieval along with all summaries
    return $bestImg, bestText, bestScore,$
      $imageSummaries, nodeSummaries$

III. EXPERIMENTAL EVALUATION

This section describes the experimental setup used to evaluate the different system components. Firstly, the dataset and action prediction tasks are described in Sec. III-A. Statistical tests used to determine significance are discussed in Sec. III-B. The evaluation metrics for RAG Image-query optimization are presented in Sec. III-C. The metrics used to evaluate the performance of different image embedding models are shown in Sec. III-D. Lastly, the metrics used for

evaluating action prediction performance and a summary of evaluated models and methods are shown in Sec. III-E.

A. Dataset and tasks

For evaluation of our approaches, we designed a sequence-prediction dataset pertaining to robotic recycling of waste electronic devices. The dataset consists of 196 images of 6 different electronic devices - 4 types of smoke detectors and 2 types of heat-cost allocators. Using this dataset, we address the following tasks: predicting the next disassembly action, estimating the number of remaining actions, listing the remaining actions, and determining whether the image depicts the final action. The dataset contains sequences of images of a variety of discarded electronic devices, showing each device in different states of disassembly. For each image, ground-truth data regarding the device name and type, next disassembly action, the last applied disassembly action, the number of remaining disassembly actions, and the list of follow-up disassembly actions. Additionally, the end-states of disassembly are labeled as such.

The previous and next step prediction encompasses 8 different robotic actions. The list of possible actions includes:

1) placing the object into a CNC machine,
2) placing the object into a clamping vise,
3) performing robotic levering on the object,
4) cutting the object within a linear pneumatic cutter,
5) performing a rectangular CNC cut,
6) performing a circular CNC cut,
7) CNC cutting of battery contacts,
8) removing a battery by wiggling.

To evaluate the RAG Image-query improvement and the performance of different embedding models, the initial dataset is used to create a labeled cropping dataset, where we labeled the electronic device along with the relevant neighboring machine tools (e.g., a clamping device), since this information provides the required context for making predictions.

B. Statistical significance testing

To evaluate whether the improvement between various methods is statistically significant, we apply McNemar's test [26], which is appropriate for paired dichotomous outcomes (each model's prediction is correct or incorrect).

C. RAG Image-query optimization

For both the grid-based and bounding-box-based methods, the following evaluation metrics were calculated after performing RAG query optimization:

- coverage c: the proportion of labeled polygon that was captured by VLM crop,
- image data removal r: the proportion of area that was removed from the initial image.

Coverage c is calculated as shown in equation below

$$c = \frac{\text{Area}(P \cap G)}{\text{Area}(G)}, \tag{2}$$

where P denotes the predicted bounding box, and G is the ground truth polygon. Image data removal r is calculated as shown in Eq. (3)

$$r = 1 - \frac{\text{Area}(I_\text{r})}{\text{Area}(I_\text{i})}, \tag{3}$$

where I_i denotes the area of the initial image, and I_r denotes the retained area of the predicted bounding box.

D. Retrieval accuracy of image embedding models

The performance of a RAG system depends on the database quality - the similarity of database examples, as compared to the input queries - and the performance of the embedding models used. The key metrics are:

1) Top-1 retrieval accuracy: The percentage of cases where the top-1 database-retrieved image matches the query image in device type and disassembly step,
2) Rank of correct result: The top-n number, where retrieved data point n matches the query image. This indicates how many RAG elements the VLM should process before finding a match in case of an inaccurate result from the image encoder.

To quantify the effectiveness of RAG image-query optimization on baseline RAG retrieval performance, we evaluate two aspects: (i) the performance of different open-source embedding models, and (ii) the effect of cropping on the performance of the embedding models, compared to vector retrieval using full-size, uncropped images.

To objectively evaluate the retrieval performance of different embedding models for both the top-1 and rank of correct results criteria, we perform tests of distribution fit on the rank of correct result histograms. The fits of exponential and log-normal distributions are evaluated using the Akaike Information Criterion (AIC) [27].

The following open-source embedding models are evaluated:

1) OpenAI CLIP small (*openai/clip-vit-base-patch32*) [8]
2) OpenAI CLIP large (*openai/clip-vit-large-patch14-336*) [8]
3) LAION CLIP large (*laion/CLIP-ViT-H-14-laion2B-s32B-b79K*) [28]
4) SigLIP2 large (*google/siglip2-large-patch16-512*) [29]

E. Action prediction

Several types of models were evaluated, including classifiers based on Convolutional Neural Networks (CNNs) and Transformers, a dummy random classifier, as well as open source and commercial VLMs. For VLMs, the system prompt always includes the list of available actions, including their preconditions and effects. Function calling is used, therefore the model is given a list of possible valid outputs. The algorithms are only required to select an action, while the parameters for each action are predicted by specific custom sub-routines.

The evaluation metrics for different tasks described in Sec. III-A are accuracy, except for the number of remaining steps, where the prediction metric is mean squared error. For

RAG-based approaches, the retrieval database contains two representative images per disassembly step per device—that is, for each of the n devices and each of their m disassembly steps, we include two representative images, rather than all available images (typically 10 per step). The remaining images are reserved for testing. Although the retrieval database and test set contain images of the same devices, the exact test images are excluded from the database to prevent leakage. During retrieval, we further enforce that only one image per disassembly step of a device is returned among the retrieval results. This step prevents multiple near-duplicates of the same step from being retrieved, encouraging the model to consider a wider range of devices and steps, thereby promoting greater diversity and generalization in RAG results.

We evaluate a variety of image classification models, along with VLMs in different settings. The following image classification models were evaluated:

- CLIP (*openai/clip-vit-base-patch32*) [8],
- ResNet-152 (*microsoft/resnet-152*) [22], and
- VisionTransformer (*google/vit-base-patch16-224*) [23].

Two open-source VLMs, along with a single commercial VLM were evaluated:

- Qwen2.5-VL-72B-Instruct with full-precision weights [30],
- Qwen3-VL-30B-A3B Mixture-of-Experts model with full precision weights [31], and
- OpenAI's GPT-4o [32].

For zero-shot prediction experiments, the models are provided with a query image showing the electronic device to be disassembled. The prompt includes:

- A general instruction to analyze the image in detail, focus on the environment (e.g., flat table, CNC machine, ...), the shape and color of the device, and any visible internal components.
- A detailed description of the available robotic workcell actions, including their preconditions and effects.
- A definition of all prediction tasks - previous step, next step, remaining steps, number of remaining steps, and whether the next step is the last step - together with the set of valid outputs for each task.
- The previously executed robotic action corresponding to the query image.

In addition to the zero-shot setting, several RAG-based variants were evaluated. In these configurations, the zero-shot prompt is augmented with additional retrieved data:

- Baseline Single-RAG
  The prompt is extended with the most similar database example retrieved using the image embedding model. The retrieved entry consists of a candidate image and its associated ground-truth QA pairs.
- Bad-Single-RAG
  A deliberately incorrect database image is randomly selected - showing either a different electronic device or a different disassembly step. As in the Single-RAG, the image and its ground-truth text are appended to the VLM prompt. This setup evaluated how much action prediction accuracy degrades when no appropriate database example exists for a novel device.
- VLM-guided RAG
  The retrieval process described in Sec. II-D is used, combining similarity information from the image embedding model with VLM reasoning over the Knowledge Tree. The highest-ranked image and its corresponding text are added to the VLM prompt, analogously to the other RAG settings. Additionally, detailed metadata from the retrieval process is recorded (e.g., image scores, VLM-generated justifications, the set of evaluated Knowledge Tree leaves, and the final selected image), enabling a separate, fine-grained analysis of the search procedure itself.

IV. Results

This section presents a comprehensive evaluation of the different parts of our VLM-Guided RAG framework. The effectiveness of VLMs in the image-query improvement step is quantified in Sec. II-B, showing that even VLMs not trained in object detection are effective in removing irrelevant image elements.

An evaluation of different open-source image embedding models and their retrieval accuracy on both original and improved (cropped) images is shown in Sec. IV-B, where it can be seen that cropping of the images results in a significant increase in retrieval accuracy. The retrieval accuracy using our VLM-Guided RAG compared to baseline image embedding models is shown in Sec. IV-C, demonstrating a statistically significant improvement.

Zero-shot action prediction results for image classification models are shown in Sec. IV-D.1. In Sec. IV-D.2, we show that fine-tuning of VLMs is not effective on our small dataset. Notably, due to computational budget constraints, only a relatively small VLM is fine-tuned.

The action prediction accuracies for zero-shot, Single RAG and VLM-guided RAG approaches are shown in Sec. IV-D.3. Since prediction latency is an important factor in real-world usability, a comparison of prediction latencies of zero-shot approaches compared to our VLM-Guided RAG is shown in Sec. IV-E. Lastly, an analysis of failure modes for various models and methods is shown in Sec. IV-F.

A. RAG Image-query optimization

The obtained results for the grid and bounding-box cropping are shown in Table I. The iterative bounding box cropping method demonstrates superior results compared to grid-based cropping, which we suspect is due to it enabling higher resolution of selecting relevant image parts. The *Qwen3-VL-30B* model demonstrates the best performance, removing **49.6%** of the image while retaining **95.4%** coverage of key image regions.

To enable fair comparison between different models that do not execute the image query optimization step all images are manually cropped for further sequence prediction tasks.

TABLE I

COVERAGE AND IMAGE DATA REMOVAL METRICS FOR VLM CROPS.

Model	Cov. (%)	Data Removal (%)
Qwen2.5-VL-72B - grid cropping	**95.85**	7.3
Qwen3-VL-30B - grid cropping	73	**44.2**
GPT-4o - grid cropping	78.5	38.4
Qwen2.5-VL-72B - bbox cropping	92.8	39.7
Qwen3-VL-30B - bbox cropping	**95.4**	**49.6**
GPT-4o - bbox cropping	81.5	**71.5**

B. Retrieval accuracy of image embedding models

Top-1 retrieval accuracy of the evaluated embedding models on both cropped images and full-size images is shown in Fig. 6. In all cases, cropping of the images improves the top-1 retrieval performance of evaluated embedding models. The relative improvements range from **31.3%** for the *OpenAI CLIP small* model, to **68.5%** for *SigLIP 2*.

On full, non-cropped images, *OpenAI CLIP Large* shows the highest top-1 accuracy of **46.88%**. On cropped images, the *SigLIP* 2 model displays the highest top-1 accuracy of **68.47%**, representing a relative improvement of **23.5%** over the smaller base *OpenAI CLIP* model. This is due to larger model size, the larger size of the training dataset, and enhancements in the model architecture [29].

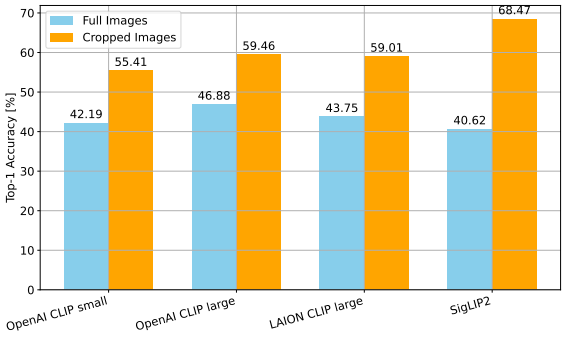


Fig. 6. Top-1 retrieval accuracy of evaluated models.

The results show that the log-normal distribution provides the closest fit. The fitted distributions for evaluated models is shown in Fig. 7. We can conclude that the *SigLIP 2* model outperforms the others both in terms of top-1 retrieval accuracy and the position of correct result, with the mean rank of correct result being **2.43** elements. This indicates that on average, one of the top-3 retrieved data points matches the query image. Therefore, for further experiments using RAG, the *SigLIP2* embedding model is used.

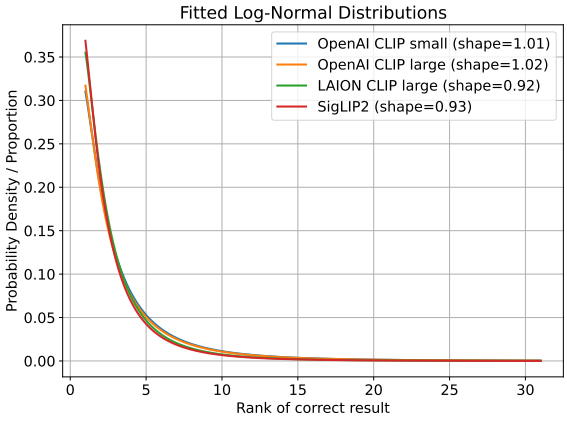


Fig. 7. Results of fitting the log-norm distribution on rank of correct result histogram.

C. Retrieval accuracy of VLM-guided RAG

Before testing action prediction accuracy, we separately evaluate the retrieval performance of VLM-Guided RAG compared to baseline RAG using the *SigLIP 2* embedding model. The resulting absolute and relative improvements for each model are shown in Table II.

The *GPT-4o* is the most accurate, at the cost of longest prediction latency due to larger model size. We observe that the *Qwen3-VL-30B* model provides the best tradeoff between reasonable search times and accuracy. This is due to its Mixture-of-Experts architecture [33], which during inference activates only a smaller subset of parameters.

TABLE II

RETRIEVAL ACCURACY COMPARISON ACROSS MODELS

Model	Accuracy (%)	Abs. Impr. (%)	Rel. Impr. (%)
Base RAG	68.47	–	–
Qwen2.5-VL-72B	84.57	16.1	23.5
Qwen3-VL	85.41	16.94	24.7
GPT-4o	**89.29**	**20.82**	**30.4**

To evaluate whether the differences in retrieval performance between different models using the VLM-guided RAG are statistically significant, we perform McNemar's test, as described in Sec. III-B. The results are shown in Table III. It can be seen that the performance difference between *Qwen2.5-VL-72B* and *Qwen3-VL-30B* is not statistically distinguishable ($p = 1.0$), while the differences between *GPT-4o* and both *Qwen2.5-VL-72B* and *Qwen3-VL-30B* are statistically significant at the 5% level ($p = 0.0158$ and $p = 0.0268$, respectively), indicating that *GPT-4o* model achieves consistently higher retrieval accuracy in the VLM-guided RAG setting.

Models	GPT-4o	Qwen2.5	Qwen3
GPT-4o	/	0.0158	0.0268
Qwen2.5	/	/	1.0

TABLE III

MCNEMAR'S TEST RESULTS COMPARING VLM-GUIDED RAG RETRIEVAL ACCURACY.

D. Action prediction & ablation study

We performed an ablation study to test the effectiveness of the VLM-guided RAG approach for next action prediction using both open-source and commercial VLMs. Firstly, we compared it to the results obtained by fine-tuning image classification models. Secondly, it was compared to fine-tuning a small VLM, and lastly, it was compared to zero-shot prediction accuracies of the VLMs.

1) Image classification models: For single task prediction, we fine-tuned several classifiers based on CNNs and Transformer-based models, as described in Sec. III-E. The algorithms are also compared to random choice, as suggested by [34]. All models were evaluated K-fold cross-validation with $K = 5$. Models were trained and evaluated on each task separately. A maximum of 30 epochs were used for training. Early stopping was implemented based on validation loss, where training was stopped when validation loss did not improve after 2 epochs. The models with the lowest validation loss were evaluated. The results are shown in in Fig. 8.

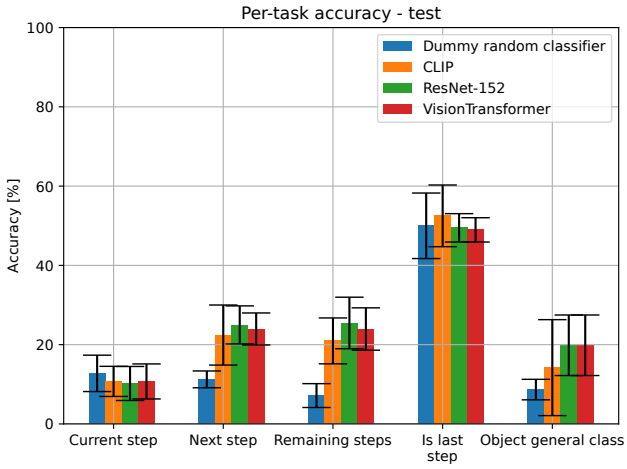


Fig. 8. Results of fine-tuning vision models.

Although the performance on the training set is high (**90%+**), the models can not generalize due to small dataset size, as well as lack of relevant textual input. The trained models outperform the random classifier.

2) VLM - fine-tuning: The results of fine-tuning a VLM are shown in Fig. **??**. The VLM was trained up to a maximum of 20 epochs, however early stopping based on validation loss was used to prevent overfitting [35], and in most cases, the training was finished by epoch 3. Fig. **??** compares the zero-shot model accuracy to the fine-tuned model. It can be seen that the accuracy significantly improves to over 90%, indicating that the model successfully improves on the training dataset, as is expected.

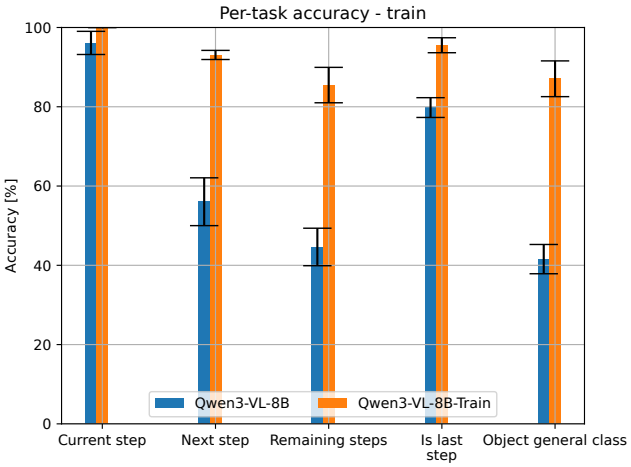


Fig. 9. Results of fine-tuning the VLM on the training set.

However, the results in Fig. 10 show that on the test set, which was left out during training, fine-tuning is not effective due to the small dataset used. Additionally, due to computational limitations, fine-tuning can not be performed on larger models, therefore a smaller model with $8B$ parameters is used, which in itself decreases action prediction accuracy.

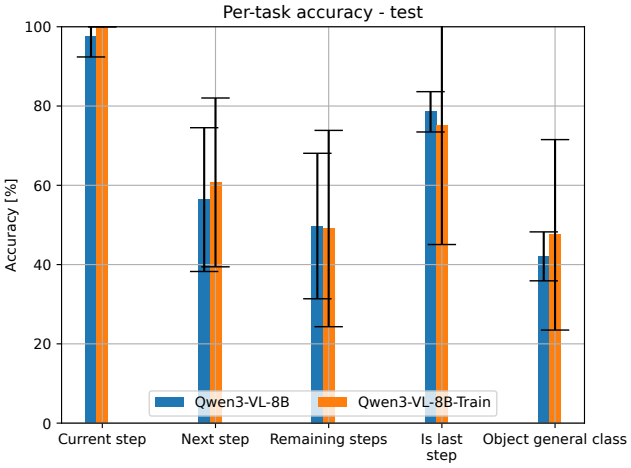


Fig. 10. Results of fine-tuning the VLM on the test set.

3) VLMs - no fine-tuning: VLMs were evaluated using a multi-task approach, i.e. predicting the outputs for all tasks in a single function call. To ensure repeatability, the generation strategy used by VLMs was greedy decoding.

In Fig. 11, the results of action prediction using VLMs for each task are shown. Compared to Fig. 8, it is clear that VLMs outperform the image classification models, such as ResNet and CLIP. This is due to the increased information content provided by the relevant text which describes the

Model	C. s. mean	C. s. std	N. s. mean [%]	N. step std	Rem. steps mean [%]	Rem. steps std	Rem. count mean [%]	Final step mean [%]	Final step std	Obj. general class mean [%]	Obj. general class std
Random classifier	9.18	1.41	11.02	1.05	7.04	0.58	**6.14**	49.69	1.46	9.18	1.80
CLIP	60.41	9.41	62.35	8.78	67.24	0.12	1.44	76.94	4.59	62.86	9.40
ResNet-152	68.47	1.84	**72.45**	1.74	69.69	0.09	0.97	**85.41**	0.83	75.10	2.49
VisionTransformer	**69.69**	2.75	**72.45**	2.12	**73.37**	0.15	0.86	84.29	1.49	**77.14**	1.89

TABLE IV

PERFORMANCE OF VARIOUS MODELS

action possibilities, as well as the knowledge gained by pre-training on large datasets containing general world knowledge.

For the zero-shot predictions, shown in Fig. 11, there is a clear difference in accuracy between the different models. The commercial *GPT-4o* model outperforms both Qwen variants, reaching 58% accuracy in next-step prediction. Nonetheless, such accuracy is insufficient for industrial use-cases. Notably, it can be seen that the *Qwen3-VL-30B* sometimes fails to correctly predict the current step, although this information is included in the prompt and only needs to be repeated by the VLM. This is likely due to the model's Mixture of Experts architecture, which activates only a small subset of parameters during inference.

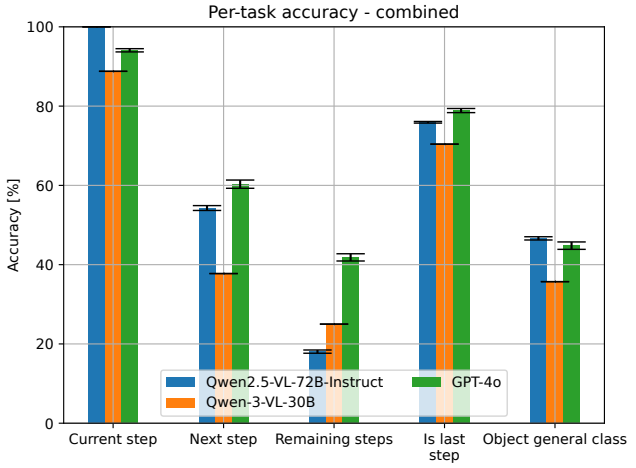


Fig. 11. Zero-shot accuracies of various models.

Compared to the zero-shot accuracy, the baseline Single-RAG approach shows significantly higher prediction accuracies shown in Fig. 12 at minimal inference latency increase (shown in Sec. IV-E). Additionally, the accuracy could be further improved by increasing the number of RAG examples in the dataset.

In the case of Bad-Single-RAG, where a VLM is deliberately given a misleading candidate image and corresponding text, it can be observed that in most cases, the VLMs are able to recognize that the candidate image is irrelevant and also that the provided corresponding text is not helpful, as shown by results in Fig. 13. In some cases, the accuracy is even higher than the zero-shot baseline. We hypothesize that the additional example helps "ground" the VLM in seeing which action is **not** applicable, thereby reducing the number of applicable actions.

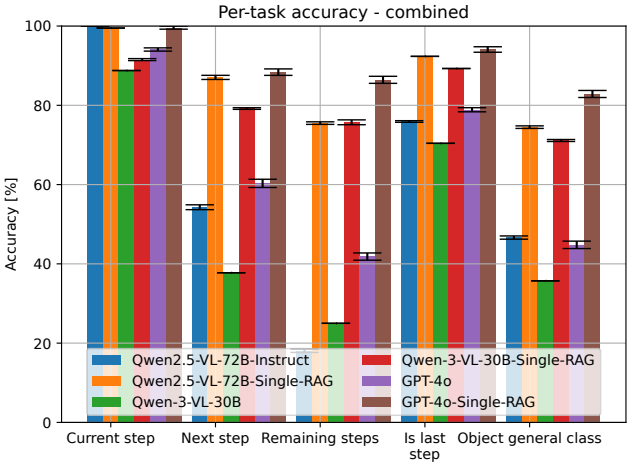


Fig. 12. Results of zero-shot vs Single-RAG.

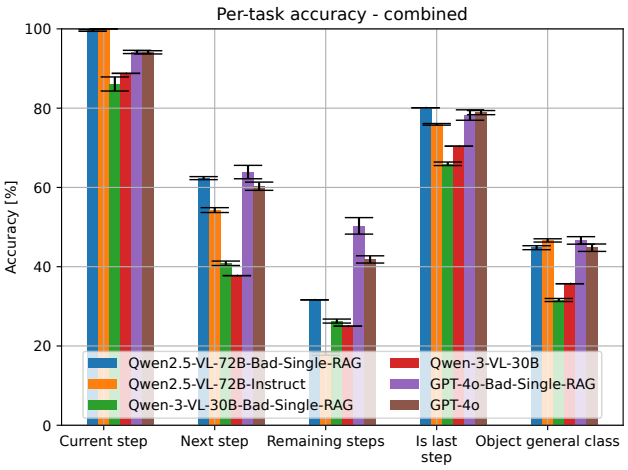


Fig. 13. Results of Bad-Single-RAG vs zero-shot.

To evaluate whether Bad-Single-RAG results are statistically distinguishable from the zero-shot variation, we perform McNemar's test. The results are shown in Table V. At the 5% significance threshold, the difference is statistically insignificant.

Compared to the Single-RAG approach, it can be seen that in all cases, VLM-guided RAG approaches exhibit increased accuracy, as shown in Fig. 14.

The difference between Single-RAG and VLM-guided RAG is shown to be statistically significant at the 0.05% threshold for all models, as shown in Table VI.

Model Family	Compared Variants	$\chi^2(1)$	p-value
Qwen2.5-VL	Zero-shot vs. Bad Single RAG	3.67	0.055
Qwen3-VL	Zero-shot vs. Bad Single RAG	0.129	0.72
GPT-4o	Zero-shot vs. Bad Single RAG	0.114	0.735

TABLE V

MCNEMAR'S TEST RESULTS COMPARING BASELINE ZERO-SHOT PREDICTIONS TO BAD SINGLE RAG.

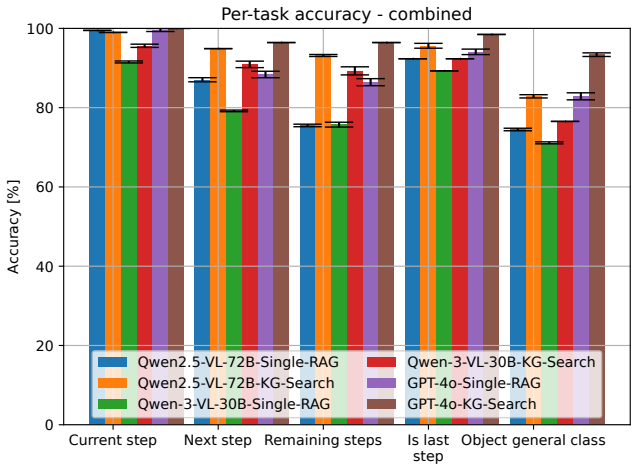


Fig. 14. Results of Single-RAG vs VLM-guided RAG.

Model Family	Compared Variants	$\chi^2(1)$	p-value
Qwen2.5-VL	Single-RAG vs. KG-Search	8.45	0.0037
Qwen3-VL	Single-RAG vs. KG-Search	9.03	0.0027
GPT-4o	Single-RAG vs. KG-Search	8.45	0.0037

TABLE VI

MCNEMAR'S TEST RESULTS COMPARING SINGLE-RAG TO VLM-GUIDED RAG.

E. Prediction latency

Prediction latency is a key criteria of system usability in real-world conditions, since industrial robots are expected to operate with minimal downtime, e.g. when waiting for action prediction results. Despite this, in robotic disassembly, action prediction accuracy is more important than only prediction latency, since incorrect predictions may lead to catastrophic results, e.g., battery fires may occur if incorrect CNC cutting methodology is utilized.

In Table VII, the mean prediction latencies and standard deviations are presented for various algorithms and cases. All results are presented for the Qwen3-VL 30B model, since it performs best in VLM-guided RAG, as well as being competitive in the zero-shot scenario. While the mean prediction latency for one image (zero-shot prediction) is **5.21 seconds**, prediction latencies for RAG-augmented cases, which require processing two images and additional image text, takes on average **6.15 s**. Prediction latency in case of VLM-guided RAG is longer, since in the best case scenario, at least 2 extra images must be processed, i.e., as the VLM must additionally confirm that the retrieved image matches the query. In this case, the mean prediction latency increases to **9.41 s**. Performing a full knowledge graph search is significantly slower, with a mean elapsed time of **157.25 s**. However, due to there being many cases where the first retrieved image is judged correct, the VLM-guided RAG search on average takes **37.7 s**.

Model	Prediction time mean [s]	Pred. t. std [s]
Zero-shot (single image)	**5.21**	**0.81**
Single RAG (two images)	6.15	0.89
KG search – Top-1 result correct	9.41	2.55
KG search – Top-1 result incorrect	157.25	86.8
KG search – average	37.7	70.3

TABLE VII

PREDICTION TIMES FOR DIFFERENT METHODS.

F. Failure modes

We perform an analysis of failure modes to determine the reasons for incorrect action prediction results. Two separate cases are considered - 1) The cases in which Single-RAG fails, 2) the cases in which VLM-guided RAG retrieval fails, and 3) the cases in which next-action prediction fails.

For the failure cases of Single-RAG, shown in Fig. 15, we observe that retrieval fails in case of subsequent disassembly steps where minimal visual difference can be observed, e.g. before and after battery contact cutting is performed within the CNC mill. This is due to the RAG retrieval step only determining similarity between images, but not using additional textual information, e.g. the previously performed step, which can additionally help to delineate and improve retrieval in the case of very similar images. Some example failure cases are shown in Fig.

In the case of VLM-guided RAG, out of 196 images, we observe only one catastrophic failure where the objects on the query and retrieved images are clearly different, shown in the center of Fig. 16. In other cases, the device types may be different, however, the images are semantically similar with same disassembly steps, which is why the next-action prediction accuracy is higher than would be suggested by the accuracy of the VLM-guided image retrieval. Additionally, most failure cases **69.2%**) occur in the case of the VLM confirming the most-similar RAG image as correct, thereby skipping the Knowledge Tree search phase. Accuracy would be higher in case of always forcing the Knowledge Tree search, but at the cost of significantly higher prediction times.

When evaluating the next action prediction errors, it can be seen that VLMs often confuse circular and rectangular CNC cutting approaches, seemingly missing the extruded rectangular battery outline sometimes present on smoke detectors. Since such a feature is easily distinguishable on depth images, future work could be aimed at enabling VLMs to also use depth information for better spatial comprehension.

G. Application in robotic workcell

The developed VLM-guided RAG approach was implemented in a modular robotic workcell [36] as a Robot

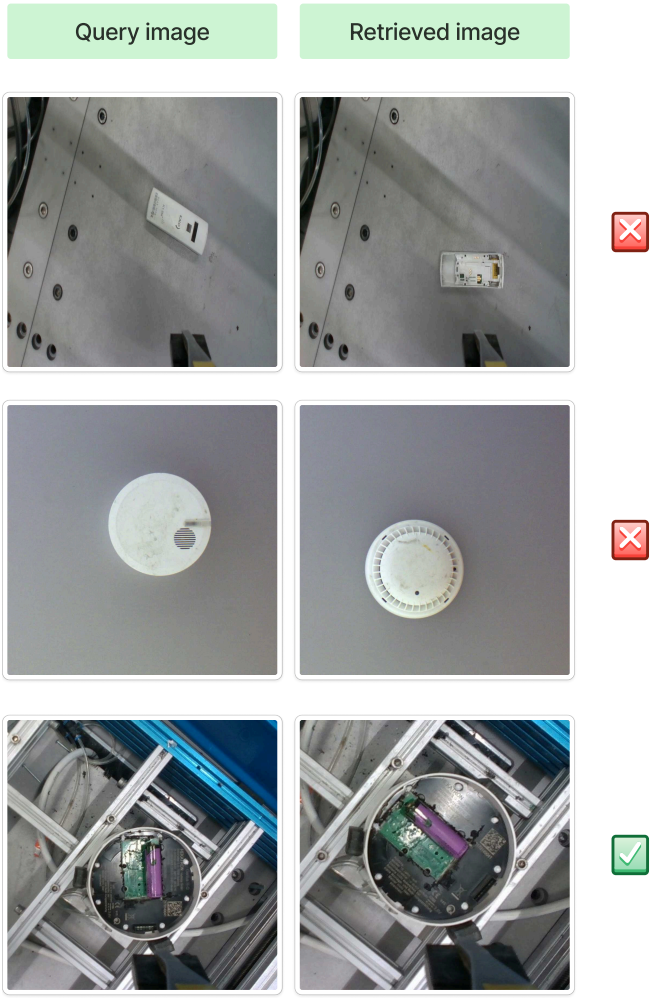


Fig. 15. Failure cases for Single RAG image retrieval.

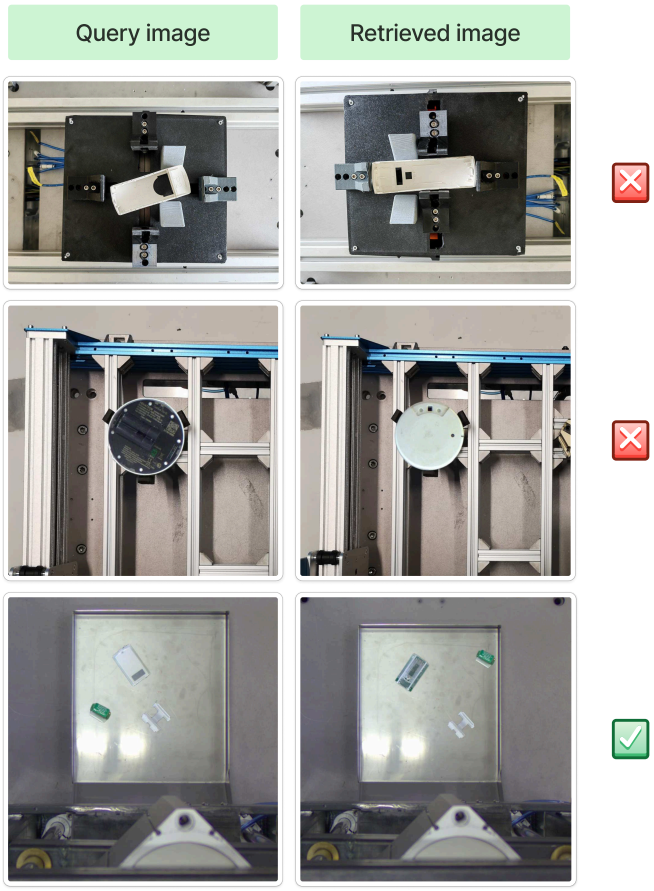


Fig. 16. Failure cases for VLM-guided RAG.

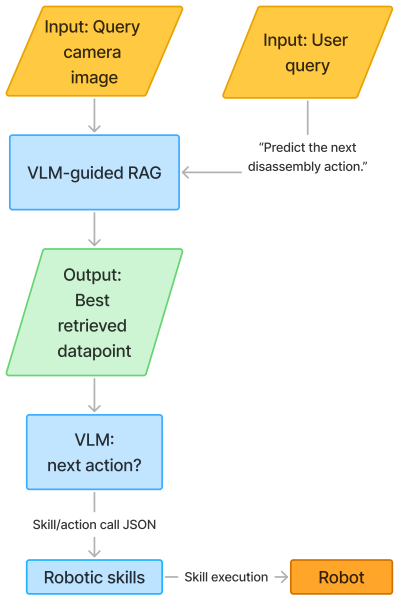


Fig. 17. The final action prediction & execution pipeline.

Operating System (ROS) node. Based on the device to be disassembled, the VLM-guided RAG module is utilized to retrieve a relevant data point and corresponding text from the database, based on the current snapshot of the workcell. The corresponding text contains the disassembly steps and the device type. This is then passed to the action prediction VLM, along with information about the available robotic disassembly skills, as shown in Fig. 17. The additional information from the VLM-guided RAG retrieved data point improves the action prediction performance of the VLM, as shown in Fig. 11. Additionally, it enables the workcell to be quickly adapted to new devices and device types, by adding required information (i.e., novel device images and disassembly steps) to the RAG database.

V. Discussion and Conclusions

We have presented an approach for robotic action prediction utilizing a VLM along with a local database and RAG,

where the VLM firstly improves the RAG input query image, then iteratively queries the database and determines the most suitable relevant images and text content. The retrieved best image and summarized text were used by the VLM in the context of next-action prediction in a robotic workcell. We compared our VLM-guided RAG approach to a baseline approach not utilizing RAG (zero-shot), an approach utilizing a single-best retrieved RAG image (Single-RAG), and an approach where the model is deliberately given a misleading image and corresponding text (Bad-Single-RAG).

Firstly, the image query improvement step was shown to be successful in improving the RAG databaase and the query image by removing unnecessary elements, while retaining the relevant electronic devices within the image crop.

Secondly, a variety of open-source image embedding models were evaluated. It was shown that the image query improvement step significantly improves the retrieval performance of the baseline RAG system. Additionally, larger embedding models exhibit both higher top-1 and top-n retrieval accuracies.

Thirdly, the zero-shot action prediction performance of both open-source and commercial models was evaluated. It was compared to the Bad-Single-RAG approach, where the models were deliberately given a misleading image, to simulate the case where the RAG database contains no relevant datapoints. It was shown that in this case, the performance of the models does not degrade.

The Single-RAG approach was shown to significantly increase the action prediction accuracy compared to zero-shot predictions. Crucially, it led to a minimal prediction latency increase. Fine-tuning was not effective on our dataset due to its small size and was outperformed by RAG-based approaches.

Lastly, the VLM-guided RAG approach was evaluated, showing an increases both in the retrieval accuracy and the final action prediction accuracy. This approach was compared to Single-RAG and showed a statistically significant increase in action prediction accuracy for all models at the cost of extending the prediction latency, since the time required to perform the VLM-guided RAG step is not deterministic.

In conclusion, the Image Query Optimization and VLM-guided RAG approaches significantly increase the operational autonomy of the robotic workcell, enabling it to be quickly adapted to disassembly of new device types without constant model fine-tuning, which was shown to be ineffective on smaller datasets, as in our case.

REFERENCES

[1] M. Simonič, B. Kuster, M. Mavsar, G. Šavle, S. Ruiz, M. Bem, M. M. Hrovat, and A. Ude, "An application study on reconfigurable robotic workcells and policy adaptation for electronic waste recycling," in *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 2024, pp. 180–186.

[2] F. Abhimanyu, T. Zodage, U. Thillaivasan, X. Lai, R. Chakwate, J. Santillan, E. Oti, M. Zhao, R. Boirum, H. Choset, and M. Travers, "Rgb-x classification for electronics sorting," 2022. [Online]. Available: https://arxiv.org/abs/2209.03509

[3] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld, "Pddl - the planning domain definition language," 08 1998.

[4] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu, "Towards generalist robot policies: What matters in building vision-language-action models," 2024. [Online]. Available: https://arxiv.org/abs/2412.14058

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[6] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.19473

[7] S. Towhidul, S. M. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das. (2024) A comprehensive survey of hallucination mitigation techniques in large language models. [Online]. Available: https://doi.org/10.48550/arXiv.2401.01313

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[9] M. Mortaheb, M. A. A. Khojastepour, S. T. Chakradhar, and S. Ulukus, "Rag-check: Evaluating multimodal retrieval augmented generation performance," 2025. [Online]. Available: https://arxiv.org/abs/2501.03995

[10] R. Lakatos, P. Pollner, A. Hajdu, and T. Joó, "Investigating the performance of retrieval-augmented generation and domain-specific fine-tuning for the development of ai-driven knowledge-based systems," *Machine Learning and Knowledge Extraction*, vol. 7, no. 1, 2025. [Online]. Available: https://www.mdpi.com/2504-4990/7/1/15

[11] T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T. R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Žiga Gosar, I. Kovač, and A. Ude, "Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells," *Robotics and Computer-Integrated Manufacturing*, vol. 66, p. 101979, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584519306726

[12] M. Mavsar, M. Denisa, B. Nemec, and A. Ude, "Intention recognition with recurrent neural networks for dynamic human-robot collaboration," in *2021 20th International Conference on Advanced Robotics (ICAR)*, Dec 2021, pp. 208–215.

[13] Z. Li, X. Wu, H. Du, F. Liu, H. Nghiem, and G. Shi, "A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges," 2025. [Online]. Available: https://arxiv.org/abs/2501.02189

[14] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," 2020.

[15] X. Zheng, Z. Weng, Y. Lyu, L. Jiang, H. Xue, B. Ren, D. Paudel, N. Sebe, L. V. Gool, and X. Hu, "Retrieval augmented generation and understanding in vision: A survey and new outlook," 2025. [Online]. Available: https://arxiv.org/abs/2503.18016

[16] Z. Gao, B. Zhang, P. Li, X. Ma, T. Yuan, Y. Fan, Y. Wu, Y. Jia, S.-C. Zhu, and Q. Li, "Multi-modal agent tuning: Building a vlm-driven agent for efficient tool usage," 2025. [Online]. Available: https://arxiv.org/abs/2412.15606

[17] B. T. Willard and R. Louf, "Efficient guided generation for large language models," *arXiv preprint arXiv:2307.09702*, 2023.

[18] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," 2022. [Online]. Available: https://arxiv.org/abs/2212.10496

[19] S. Yu, C. Tang, B. Xu, J. Cui, J. Ran, Y. Yan, Z. Liu, S. Wang, X. Han, Z. Liu, and M. Sun, "VisRAG: Vision-based retrieval-augmented generation on multi-modality documents," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=zG459X3Xge

[20] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.

[21] X. Lei, Z. Yang, X. Chen, P. Li, and Y. Liu, "Scaffolding coordinates to promote vision-language coordination in large multi-modal models," 2024. [Online]. Available: https://arxiv.org/abs/2402.12058

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[24] Y. Du, M. Tian, S. Ronanki, S. Rongali, S. Bodapati, A. Galstyan, A. Wells, R. Schwartz, E. A. Huerta, and H. Peng, "Context length alone hurts llm performance despite perfect retrieval," 2025. [Online]. Available: https://arxiv.org/abs/2510.05381

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[26] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947. [Online]. Available: https://doi.org/10.1007/bf02295996

[27] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[28] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," 2021. [Online]. Available: https://arxiv.org/abs/2111.02114

[29] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," 2025. [Online]. Available: https://arxiv.org/abs/2502.14786

[30] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," 2025.

[31] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

[32] O. et al., "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[33] M. Artetxe, S. Bhosale, N. Goyal, T. Mihaylov, M. Ott, S. Shleifer, X. V. Lin, J. Du, S. Iyer, R. Pasunuru, G. Anantharaman, X. Li, S. Chen, H. Akin, M. Baines, L. Martin, X. Zhou, P. S. Koura, B. O'Horo, J. Wang, L. Zettlemoyer, M. Diab, Z. Kozareva, and V. Stoyanov, "Efficient large scale language modeling with mixtures of experts," 2022. [Online]. Available: https://arxiv.org/abs/2112.10684

[34] J. Metz, N. Spolaôr, E. A. Cherman, and M. C. Monard, "Comparing published multi-label classifier performance measures to the ones obtained by a simple multi-label baseline classifier," 2015. [Online]. Available: https://arxiv.org/abs/1503.06952

[35] L. Prechelt, "Early stopping - but when?" *Lecture Notes in Computer Science*, 03 2000.

[36] M. Simonič, R. Pahič, T. Gašpar, S. Abdolshah, S. Haddadin, M. G. Catalano, F. Wörgötter, and A. Ude, "Modular ROS-based software architecture for reconfigurable, Industry 4.0 compatible robotic workcells," in *20th International Conference on Advanced Robotics (ICAR)*, Ljubljana, Slovenia, 2021, pp. 44–51.

In-Person: Oral ☒ / Poster ☒ / The same ☐
Virtual: Zoom ☐ / Pre-recorded video ☐

Topic: *Machine Learning and Artificial Intelligence in Manufacturing*

Zero-Shot Benchmarking of Vision-Language Models for Device Disassembly

Fatima Aziz[1], Nikola Marić[1,2], Boris Kuster[1,2], Aleš Ude[1], and Boshko Koloski[1,2]
[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
E-mail: {fatima.aziz, nikola.maric}@ijs.si

Summary: Safe robotic disassembly of electronic waste, essential for preventing battery fires during recycling, requires sequential perception to predict current disassembly states, subsequent actions, and task completion for highly variable device types. Pre-programmed robotic scripts are impractical due to device heterogeneity, while traditional fine-tuning approaches struggle with the small datasets typical in recycling applications. Vision-Language Models (VLMs) offer promising zero-shot reasoning capabilities, but the optimal prompting strategy for multi-target perception remains unexplored: should perception tasks (previous-step, next-step, remaining-steps, last-step) be queried independently (single-task) or unified in one prompt (multi-task)? We evaluate both strategies across four open-source VLMs (LLaVA-34B, Llama3.2-Vision-90B, Gemma3-27B, Qwen2.5-72B) using six devices and 196 annotated disassembly stages across 10 independent runs per model. Multi-task prompting achieves 17.95 percentage points higher accuracy than single-task approaches, with the strongest gains in next-step and last-step predictions. These findings provide practical guidance for implementing VLM-based perception in zero-shot robotic disassembly systems where fine-tuning is infeasible.

Keywords: zero-shot VLM evaluation, vision-language models, device disassembly, prompting strategies, industrial robotics.

1. Introduction

Vision-Language models are increasingly applied to robotic automation tasks, such as electronic waste disassembly, where the wide variability of device structures prevents the application of fixed, pre-programmed scripts [1]. These applications require sequential perception, identifying the current state, predicting the next action, and estimating remaining steps [2]. Since task fine-tuning requires large datasets and sufficient compute, both of which are expensive and difficult to acquire, zero-shot evaluation without fine-tuning models remains an effective alternative relying on the model's strong generalization capabilities [3], [4], [5]. While prompt tuning and engineering of such VLMs has been studied for general vision tasks, a systematic comparison of prompting strategies for sequential disassembly remains largely unexplored [6]. Our work addresses this gap through a comparative evaluation of two distinct approaches: single-task prompting, and multi-task prompting. We analyze the trade-offs between these strategies to provide practical guidance for implementing VLM-based perception in robotic disassembly tasks. We evaluate four open-source VLMs LLaVA, Llama-3.2-Vision, Gemma-3, Qwen-2.5 to assess prompting strategy generalization across model scales and architectures.

2. Methodology

We evaluate VLMs on four tasks for device disassembly, comparing two prompting strategies that frame each task as a classification problem.

2.1. Dataset and Tasks

The dataset contains 196 images from six electronic waste devices: four types of smoke detectors and two types of heat-cost allocators. Each image represents a distinct disassembly stage with corresponding metadata including device name, type, previous action, next action, the list of remaining steps, and a flag indicating task completion.
Based on this dataset, we define four perception tasks:
Previous-Step Prediction: Predicts which prior step produced the current state shown in the image.
Next-Step Prediction: Predicts the next action in the disassembly sequence.
Remaining-Steps Prediction: Predicts the steps left to complete the disassembly.
Last-Step Prediction: Predicts whether the current state represents task completion.
These tasks test how effectively models interpret procedural visual data, anticipate future actions, and reason about process completion.

2.2. Prompting Strategies

We compare two distinct prompting methods:
Single-task prompting issues independent prompts for each of the four tasks. Each prompt receives the device name, ordered step list of possible disassembly steps for that device, and an image of the current state of the device. The model returns predictions in JSON format.
Multi-task prompting issues a unified prompt to request all four predictions at once. The prompt includes the device name, the ordered list of possible disassembly steps, and the image of the current state. The model

returns a single JSON with all predictions. This unified prompt exploits cross-task dependencies.

2.3. Models and Evaluation Setup

The evaluation is conducted on a per-image basis. For each of the 196 unique disassembly stages, models are prompted in a zero-shot setting to generate predictions for all four defined tasks. We assess four 4-bit quantized (Q4) VLMs: LLaVA 34B, Llama-3.2-Vision 90B, Gemma-3 27B, and Qwen-2.5 72B, all deployed locally on our institute server equipped with four NVIDIA RTX 4090 24GB GPUs. Gemma-3 requires one GPU when loaded, LLaVA occupies two GPUs, while Llama and Qwen utilize all four GPUs. Each model is tested with both prompting strategies across the full dataset using ten independent runs with identical configurations. Model outputs are parsed from JSON responses, and each task is evaluated in terms of accuracy. A dummy classifier serving as a baseline is also evaluated; this classifier randomly selects predictions for all tasks from the set of possible disassembly steps.

3. Results and Discussion

The evaluation of single-task (ST) versus multi-task (MT) prompting, detailed in **Table 1**, reveals that a unified prompt structure is superior for zero-shot device disassembly tasks. Multi-task prompting significantly improves overall performance, with an average accuracy increase of 17.95 percentage points. This is attributed to the model's ability to leverage inter-task relationships for more consistent reasoning. Notably, model performance is not dictated by parameter count. The smallest model, Gemma3-27B, achieved the highest accuracy (65.42%) with MT prompting, almost doubling its ST performance. In contrast, the largest model, Llama3.2-Vision-90B performed worse than the baseline dummy classifier. Task-specific analysis highlights a clear hierarchy in difficulty. "Last-Step Detection" proved to be the most straightforward task, with Gemma3-27B and Qwen2.5-72B achieving near-perfect accuracy (>99%) with multi-task prompts. Conversely, "Remaining-Steps Prediction" was the most challenging, with LLaVA and Llama performing worse than the baseline. This indicates that some architectures may struggle with the complexity of this task, which is further complicated by the multi-task approach.

4. Conclusions

Our evaluation shows that multi-task prompting enables VLMs to exploit cross-task dependencies, resulting in more consistent sequential reasoning and higher accuracy than single-task prompting for robotic disassembly. These findings provide practical guidance for prompt design for applying VLMs to robotic disassembly and establish a reproducible baseline for future research. Future work will include expanding to a diverse dataset and deploying the highest-performing model into an industrial robotic system for real-time evaluation.

Acknowledgements

This work was supported by DIGITOP, GA no. TN-06-0106, funded by the Ministry of Higher Education, Science and Innovation of Slovenia, Slovenian Research and Innovation Agency, and European Union – NextGenerationEU. It was supported also by financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103). Young researcher grants no. PR-12394 and PR-13689 funded the work of Boshko Koloski and Nikola Marić respectively, as well as grant no. PR-12379.

Table 1. Comparison of single-task (ST) vs. multi-task (MT) prompting accuracy (%), averaged over 10 runs for four disassembly perception tasks. Values shown as mean (SD).

	Previous step		**Next Step**		**Remaining Steps**		**Last Step**		**Average**	
Models	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**	**ST**	**MT**
Gemma-3 27B	17.39 (0.00)	**54.64 (1.21)**	25.47 (1.52)	**53.86 (1.24)**	18.63 (1.97)	**53.86 (1.24)**	73.91 (0.00)	99.32 (0.25)	**33.85**	**65.42**
LLaVA 34B	23.48 (4.43)	31.97 (2.75)	5.65 (5.85)	32.51 (2.46)	21.74 (3.37)	17.64 (2.22)	63.48 (7.58)	76.80 (2.32)	28.59	39.73
Llama-3.2-Vision 90B	24.78 (5.52)	23.98 (3.12)	7.39 (3.91)	18.15 (3.14)	24.35 (3.48)	18.29 (3.95)	62.61 (9.16)	52.61 (5.18)	29.78	28.26
Qwen-2.5 72B	24.78 (4.78)	43.77 (1.66)	19.56 (5.91)	43.77 (1.66)	26.09 (0.00)	43.77 (1.66)	51.30 (5.07)	**99.92 (0.23)**	30.43	57.81
Dummy Classifier Baseline	20.57 (1.84)		20.51 (1.84)		35.38 (2.97)		50.52 (2.94)		31.75	

References

[1] S. Ameur, M. Tabaa, Z. Hidila, M. Hamlich, K. Karboub, and R. Bearee, 'The future of robotic disassembly: a systematic review of techniques and applications in the age of AI', *Front. Robot. AI*, vol. 12, Oct. 2025, doi: 10.3389/frobt.2025.1584657.

[2] J. Cui, C. Yang, J. Zhang, S. Tian, J. Liu, and W. Xu, 'Robotic disassembly sequence planning considering parts failure features', *IET Collab. Intell. Manuf.*, vol. 5, no. 1, p. e12074, 2023, doi: 10.1049/cim2.12074.

[3] Z. Wang, R. Shen, and B. C. Stadie, 'Solving Robotics Problems in Zero-Shot with Vision-Language Models', Oct. 2024, Accessed: Nov. 11, 2025. [Online]. Available: https://openreview.net/forum?id=RQDuFF1rOn

[4] A. Gupta, P. Vuillecard, A. Farkhondeh, and J.-M. Odobez, 'Exploring the Zero-Shot Capabilities of Vision-Language Models for Improving Gaze Following', in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2024, pp. 615–624. doi: 10.1109/CVPRW63382.2024.00066.

[5] Y. Li, P. Jiang, C. Chai, X. Zhang, and C. Liu, 'A framework for robotic manipulation tasks based on multiple zero shot models', *Sci. Rep.*, vol. 15, no. 1, p. 31141, Aug. 2025, doi: 10.1038/s41598-025-17015-z.

[6] S. Shen *et al.*, 'Multitask Vision-Language Prompt Tuning', presented at the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 5656–5667. Accessed: Nov. 11, 2025. [Online]. Available: https://openaccess.thecvf.com/content/WACV2024/html/Shen_Multitask_Vision-Language_Prompt_Tuning_WACV_2024_paper.html