

Mount the drive

```
from google.colab import drive  
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Create and Validate a BN

Ensure that all species dichotomous key sets actually map to the species we intend them to

```
# CONFIGS  
  
# network dependencies  
!pip install pgmpy  
from pgmpy.models import DiscreteBayesianNetwork  
from pgmpy.estimators import BayesianEstimator  
from pgmpy.inference import VariableElimination  
  
# loading dichotomous key data  
import json  
import os  
import pandas as pd  
import ast  
import numpy as np  
  
Collecting pgmpy  
  Downloading pgmpy-1.0.0-py3-none-any.whl.metadata (9.4 kB)  
Requirement already satisfied: networkx in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (3.6)  
Requirement already satisfied: numpy in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (2.0.2)  
Requirement already satisfied: scipy in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (1.16.3)  
Requirement already satisfied: scikit-learn in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (1.6.1)  
Requirement already satisfied: pandas in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (2.2.2)  
Requirement already satisfied: torch in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (2.9.0+cu126)  
Requirement already satisfied: statsmodels in  
/usr/local/lib/python3.12/dist-packages (from pgmpy) (0.14.5)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-  
packages (from pgmpy) (4.67.1)  
Requirement already satisfied: joblib in
```

```
/usr/local/lib/python3.12/dist-packages (from pgmpy) (1.5.2)
Requirement already satisfied: opt-einsum in
/usr/local/lib/python3.12/dist-packages (from pgmpy) (3.4.0)
Collecting pyro-ppl (from pgmpy)
  Downloading pyro_ppl-1.9.1-py3-none-any.whl.metadata (7.8 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.12/dist-packages (from pandas->pgmpy)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.12/dist-packages (from pandas->pgmpy) (2025.2)
Collecting pyro-api>=0.1.1 (from pyro-ppl->pgmpy)
  Downloading pyro_api-0.1.2-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.20.0)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (4.15.0)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (1.14.0)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.1.6)
Requirement already satisfied: fsspec>=0.8.5 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy)
(9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy)
(10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.5.4.2)
Requirement already satisfied: nvidia-cusparseelt-cu12==0.7.1 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in
```

```

/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (2.27.5)
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.3.20)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (1.11.1.6)
Requirement already satisfied: triton==3.5.0 in
/usr/local/lib/python3.12/dist-packages (from torch->pgmpy) (3.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.12/dist-packages (from scikit-learn->pgmpy)
(3.6.0)
Requirement already satisfied: patsy>=0.5.6 in
/usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy)
(1.0.2)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.12/dist-packages (from statsmodels->pgmpy)
(25.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2-
>pandas->pgmpy) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch-
>pgmpy) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->torch->pgmpy)
(3.0.3)
Downloading pgmpy-1.0.0-py3-none-any.whl (2.0 MB)
████████████████████████████████████████████████████████████████████████ 2.0/2.0 MB 33.0 MB/s eta
0:00:00
████████████████████████████████████████████████████████████████████████ 756.0/756.0 kB 47.5 MB/s eta
0:00:00
py
Successfully installed pgmpy-1.0.0 pyro-api-0.1.2 pyro-ppl-1.9.1

# DIRS
ROOT = '/content/drive/MyDrive/Semester Project'
KEYS_DIR = os.path.join(ROOT, 'data', 'keys.csv')
DICHT_DIR = os.path.join(ROOT, 'data', 'feature_names.json')

# DATA
df = pd.read_csv(KEYS_DIR, encoding="latin1")                      # Ana's data on
dichotomous keys
dicht = json.load(open(DICHT_DIR, "r"))                            # the feature
names (key names)

# HELPERS
def parse_and_fix(x):

```

```

"""Parse Ana's stored vector reliably and convert:
    1 -> 1 (present)
    0 -> 0 (absent)
    anything else (nan, None, '', etc.) -> 2 (not evaluated)
"""
# First, safely parse the nested string structure
try:
    parsed = ast.literal_eval(x)
    parsed = ast.literal_eval(parsed)
except Exception:
    # fallback: try more normalization
    cleaned = x.replace("nan", "None").replace("NaN", "None")
    parsed = ast.literal_eval(ast.literal_eval(cleaned))

vec = []
for v in parsed:
    if v in [1, "1", True]:
        vec.append(1)
    elif v in [0, "0", False]:
        vec.append(0)
    else:
        vec.append(2) # NOT EVALUATED
return np.array(vec, dtype=int)

def vector_is_valid(vec):
    if vec is None:
        return False
    if isinstance(vec, float) and np.isnan(vec):
        return False
    arr = np.array(vec, dtype=float)
    return not np.isnan(arr).any()

def cpd_to_df(cpd):
    species = list(cpd.state_names["Species"])
    df = pd.DataFrame(cpd.values,
                       index=cpd.state_names[cpd.variable],
                       columns=species)
    if len(df.index) == 2:
        df.index = ["No", "Yes"]
    return df

# validate these mappings
def predict_species(BN_model, evidence_dict, verbose=True):
    """
    Run BN inference given a feature dictionary (evidence).
    Returns (predicted_species, probability, full_distribution).
    """
    # set up inference engine

```

```

inference = VariableElimination(BN_model)

# run query
query_result = inference.query(
    variables=["Species"],
    evidence=evidence_dict
)

# extract names and probs
species_names = query_result.state_names["Species"]
probs = query_result.values

# most probable species
max_idx = probs.argmax()
predicted_species = species_names[max_idx]
predicted_prob = probs[max_idx]

if verbose:
    print("\n==== Bayesian Network Prediction ===")
    print("Predicted species:", predicted_species)
    print("Probability:", predicted_prob)
    print("\nFull distribution:")
    for sp, p in zip(species_names, probs):
        print(f" {sp}: {p:.4f}")

return predicted_species, predicted_prob, (species_names, probs)

import matplotlib.pyplot as plt

def plot_posterior(species, probs, top_k=15):
    """
    Plot the posterior distribution from the BN.
    Optionally show only the top_k species for clarity.
    """

    # Sort species by probability
    idx = probs.argsort()[:-1]          # descending
    species_sorted = [species[i] for i in idx]
    probs_sorted = probs[idx]

    # Optionally show only top-k most likely species
    species_plot = species_sorted[:top_k]
    probs_plot = probs_sorted[:top_k]

    plt.figure(figsize=(10, 5))
    plt.bar(species_plot, probs_plot)
    # plt.xticks(rotation=45, ha='right')
    plt.ylim(0, 1)
    plt.ylabel("Posterior Probability")

```

```

plt.title("Bayesian Network Species Posterior (Top {} Species)".format(top_k))
plt.tight_layout()
plt.show()

# clean the key vector
df["Full vector"] = df["Full vector"].apply(parse_and_fix)

# clean names and drop dupes
df["Species"] = df["Species"].astype(str).str.strip()
df = df.drop_duplicates(subset="Species", keep="first")
df = df[df["Full vector"].apply(vector_is_valid)]

# fill out key value pairs : (species, json of feature names)
keys = {}

for species, row in df.iterrows():
    vec = row["Full vector"]

    # Length check
    if len(vec) != len(dicht):
        print(f"ERROR: Species '{row['Species']}' has vector length {len(vec)} (expected {len(dicht)})")
        continue

    # Build feature dictionary
    feats = {feat: int(val) for feat, val in zip(dicht, vec)}
    keys[row["Species"]] = feats

from collections import defaultdict

# CHECK FOR DUPES
def build_equivalence_dict(keys):
    """
    Input: keys = { species_name : {feature_name: value, ...}, ... }
    Output: equivalence = { species_name : [species_with_same_vector], ... }
    """
    reverse = defaultdict(list)
    for sp, feadict in keys.items():
        vec_tuple = tuple(featdict.values())    # must be hashable
        reverse[vec_tuple].append(sp)

    equivalence = {}
    for vec, species_list in reverse.items():
        # all species in this group map to the same species_list
        equivalence[vec] = species_list

```

```

        for sp in species_list:
            equivalence[sp] = species_list

    return equivalence, reverse

# === RUN IT ===
equivalence, reverse_groups = build_equivalence_dict(keys)

# Print duplicate groups nicely
print("\n== Duplicate Feature Groups ==")
for vec, group in reverse_groups.items():
    if len(group) > 1:
        print(group)

# Example lookup
print("\nmarshallii group:", equivalence.get("marshallii"))
print("gambiae group:", equivalence.get("gambiae", ["gambiae"]))

== Duplicate Feature Groups ==
['marshallii', 'hughi', 'letabensis', 'kosiensis']
['funestus', 'sergentii', 'cameroni']
['listeri', 'seretsei', 'azevedoi']
['somalicus', 'nili']
['argenteolobatus', 'murphyi']
['vanhoofi', 'hamoni']
['cydippis', 'squamosus']
['coustani', 'crypticus']

marshallii group: ['marshallii', 'hughi', 'letabensis', 'kosiensis']
gambiae group: ['gambiae']

print(equivalence.get("ardensis"))

['ardensis']

# build a network
rows = []
for species, featdict in keys.items():
    row = featdict.copy()
    row["Species"] = species
    rows.append(row)

df_bn = pd.DataFrame(rows)

def expand_species(df, n=100):
    expanded = []
    for _, row in df.iterrows():
        for _ in range(n):
            expanded.append(row.copy())
    return pd.DataFrame(expanded)

```

```

df_train = expand_species(df_bn, n=100)

# build model
from pgmpy.models import DiscreteBayesianNetwork
features = [c for c in df_train.columns if c != "Species"]
edges = [("Species", feat) for feat in features]

BN_model = DiscreteBayesianNetwork(edges)

# fit to cpd
from pgmpy.estimators import BayesianEstimator
BN_model.fit(df_train, estimator=BayesianEstimator, prior_type="BDeu",
equivalent_sample_size=10)

<pgmpy.models.DiscreteBayesianNetwork.DiscreteBayesianNetwork at
0x7f61403d1df0>

# validate the model with equivalence groups
validated = True
for species, featdict in keys.items():
    evidence = featdict.copy()
    pred, _, _ = predict_species(BN_model, evidence, verbose=False)

    allowed = equivalence.get(species, [species])
    if pred not in allowed:
        print(f"ERROR: Species '{species}' predicted as '{pred}'")
        validated = False

if validated:
    print("\n==== VALIDATION SUCCESSFUL ====")

==== VALIDATION SUCCESSFUL ====

```

Load Dataset

```

# load existing dataset from /processed
from datasets import load_from_disk

drive.mount('/content/drive')
DATASET_PATH = "/content/drive/MyDrive/Semester Project/data-augmentation/processed"
dataset = load_from_disk(DATASET_PATH)

Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).

```

```

# %%capture
# import re
# if "COLAB_" not in "".join(os.environ.keys()):
#     !pip install unsloth
# else:
#     # Do this only in Colab notebooks! Otherwise use pip install
#     unsloth
#     import torch; v = re.match(r"[0-9\.]{3,}", str(torch.__version__)).group(0)
#     xformers = "xformers==" + ("0.0.32.post2" if v == "2.8.0" else
#     "0.0.29.post3")
#     !pip install --no-deps bitsandbytes accelerate {xformers} peft
#     trl triton cut_cross_entropy unsloth_zoo
#     !pip install sentencepiece protobuf "datasets>=3.4.1,<4.0.0"
#     "huggingface_hub>=0.34.0" hf_transfer
#     !pip install --no-deps unsloth
#     !pip install transformers==4.56.2
#     !pip install --no-deps trl==0.22.2

%%capture
import os, re
if "COLAB_" not in "".join(os.environ.keys()):
    !pip install unsloth
else:
    # Do this only in Colab notebooks! Otherwise use pip install
    unsloth
    import torch; v = re.match(r"[0-9]{1,}\.[0-9]{1,}", str(torch.__version__)).group(0)
    xformers = "xformers==" + ("0.0.33.post1" if v=="2.9" else
    "0.0.32.post2" if v=="2.8" else "0.0.29.post3")
    !pip install --no-deps bitsandbytes accelerate {xformers} peft trl
    triton cut_cross_entropy unsloth_zoo
    !pip install sentencepiece protobuf "datasets==4.3.0"
    "huggingface_hub>=0.34.0" hf_transfer
    !pip install --no-deps unsloth
    !pip install transformers==4.56.2
    !pip install --no-deps trl==0.22.2

from unsloth import FastVisionModel
from peft import PeftModel
import torch

# 1. Load base model
base_model, tokenizer = FastVisionModel.from_pretrained(
    "unsloth/Llama-3.2-11B-Vision-Instruct",
    load_in_4bit=True,
    dtype=torch.float16,
    use_gradient_checkpointing="unsloth",
)

```

```

# 2. Attach your LoRA adapters
lora_path = "/content/drive/MyDrive/Semester Project/data-
augmentation/lora_model"
model = PeftModel.from_pretrained(base_model, lora_path)

[] Unsloth: Will patch your computer to enable 2x faster free
finetuning.
[] Unsloth Zoo will now patch everything to make training faster!
==((=====))== Unsloth 2025.11.6: Fast Mllama patching. Transformers:
4.56.2.
    \\ /| Tesla T4. Num GPUs = 1. Max memory: 14.741 GB. Platform:
Linux.
0^0/ \_/\ Torch: 2.9.0+cu126. CUDA: 7.5. CUDA Toolkit: 12.6.
Triton: 3.5.0
\      / Bfloat16 = FALSE. FA [Xformers = 0.0.33.post1. FA2 =
False]
"-_____" Free license: http://github.com/unslotha/unsloth
Unsloth: Fast downloading is enabled - ignore downloading bars which
are red colored!

{"model_id": "0f20a0f6012845a1bcf858ef0dea7d38", "version_major": 2, "vers
ion_minor": 0}

{"model_id": "b64df841ae3749bc8007f1da75421120", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "33d5cbb8e75343538b42dfdb560fef9", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "8a2607c692634fe4bae39fe631fa9145", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "25515577d19c4aa88a381f42862b958d", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "4ae2f6c37819482f970193af566affac", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "de927a578ccb40e5a6fa8d17aa83fb85", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "41700ab3656d4cc09b83c74aad38a644", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "4fac004bf1b747639900a589cc42ab01", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "0cc732b948314759be5bcf0adbb999b9", "version_major": 2, "vers
ion_minor": 0}

 {"model_id": "5431d0e26e904854b2419a5590f88343", "version_major": 2, "vers
ion_minor": 0}

```

```

# load the basic key map (to map machine keys -> dichotomous keys)
KEYS_PATH = "/content/drive/MyDrive/Semester Project/data-
augmentation/basic_key_map.json"

with open(KEYS_PATH, "r") as f:
    basic_key_map = json.load(f)

# set up GPT client
!pip install openai
from openai import OpenAI

# set up the api key
os.environ["OPENAI_API_KEY"] = "sk-proj-KysyPg2ReTwpfZIrUb__Ut07Ia-
jKeinC5-DyQETmGsXc7s_v1jKsyJeMXL0-
BMCY78EqP9NcqT3BlbkFJc9LIWEauhyhUzyT0NmJHbb9FiKiZQuepoj0Rk00efYn-Vdo-
Dui34-SuRKETJ1h7VsFTclCXEA"
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))

Requirement already satisfied: openai in
/usr/local/lib/python3.12/dist-packages (2.8.1)
Requirement already satisfied: anyio<5,>=3.5.0 in
/usr/local/lib/python3.12/dist-packages (from openai) (4.11.0)
Requirement already satisfied: distro<2,>=1.7.0 in
/usr/local/lib/python3.12/dist-packages (from openai) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in
/usr/local/lib/python3.12/dist-packages (from openai) (0.28.1)
Requirement already satisfied: jiter<1,>=0.10.0 in
/usr/local/lib/python3.12/dist-packages (from openai) (0.12.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in
/usr/local/lib/python3.12/dist-packages (from openai) (2.12.3)
Requirement already satisfied: sniffio in
/usr/local/lib/python3.12/dist-packages (from openai) (1.3.1)
Requirement already satisfied: tqdm>4 in
/usr/local/lib/python3.12/dist-packages (from openai) (4.67.1)
Requirement already satisfied: typing-extensions<5,>=4.11 in
/usr/local/lib/python3.12/dist-packages (from openai) (4.15.0)
Requirement already satisfied: idna>=2.8 in
/usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.5.0->openai)
(3.11)
Requirement already satisfied: certifi in
/usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0-
>openai) (2025.11.12)
Requirement already satisfied: httpcore==1.* in
/usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0-
>openai) (1.0.9)
Requirement already satisfied: h11>=0.16 in
/usr/local/lib/python3.12/dist-packages (from httpcore==1.*-
>httpx<1,>=0.23.0->openai) (0.16.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0-

```

```

>openai) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0->openai) (2.41.4)
Requirement already satisfied: typing-inspection>=0.4.2 in
/usr/local/lib/python3.12/dist-packages (from pydantic<3,>=1.9.0->openai) (0.4.2)

# post-processing GPT HELPER

allowed_keys = basic_key_map.keys()

# instruction:
def enforce_json_structure(raw_text, allowed_keys):
    allowed_keys = list(allowed_keys)

    prompt = f"""
You are a strict JSON validator.

Your ONLY tasks are:
1. Ensure the output is valid JSON.
2. Ensure the JSON contains exactly this set of keys:
{json.dumps(allowed_keys, indent=2)}
3. For any key already in the input with value "present", "absent", or
"occluded",
    DO NOT MODIFY ITS VALUE.
4. If a key is missing, add it with value "occluded".
5. If a key has an invalid value, replace it with "occluded".
6. Return ONLY a JSON object. No comments, no explanations.

Input:
{raw_text}
"""

    response = client.chat.completions.create(
        model="gpt-4.1-mini",
        response_format={"type": "json_object"},
        messages=[{"role": "user", "content": prompt}],
    )

    return json.loads(response.choices[0].message.content)

# decode the json output by mapping the basic keys back to the
dichotomous keys
# also map present -> 1
#           absent  -> 0
#           else     -> exclude from json

def decode_feature_json(model_json, key_map):

```

```

decoded = {}
all_basic_keys = set(key_map.keys())

# 1. Ensure no missing/extra keys
if set(model_json.keys()) != all_basic_keys:
    missing = all_basic_keys - set(model_json.keys())
    extra   = set(model_json.keys()) - all_basic_keys
    raise ValueError(f"Key mismatch. Missing={missing},
Extra={extra}")

for basic_key, value in model_json.items():
    if basic_key not in key_map:
        raise KeyError(f"Unexpected key '{basic_key}' in
model_json")

    v = value.strip().lower()

    # 2. Strict validity
    if v not in ("present", "absent", "occluded"):
        raise ValueError(f"Invalid value '{value}' for key
'{basic_key}'")

    # 3. Map with NO silent fallback
    if v == "present":
        decoded[key_map[basic_key]] = 1
    elif v == "absent":
        decoded[key_map[basic_key]] = 0
    else:
        decoded[key_map[basic_key]] = 2

return decoded

```

Perform inference with Model

```

# enable inference mode!
FastVisionModel.for_inference(model)

# select an image
im = 250

# select an image to infer from
image = dataset[im]["image"]

# instruction to the VLM
instruction = (
    "You are an expert entomologist. Describe accurately what you see
in this "
    "image based on mosquito descriptions present in dichotomous

```

```

keys ."
)

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": instruction}
    ]}
]

input_text = tokenizer.apply_chat_template(messages,
add_generation_prompt=True)

inputs = tokenizer(
    image,
    input_text,
    add_special_tokens=False,
    return_tensors="pt",
).to("cuda")

# perform inference
outputs = model.generate(
    **inputs,
    max_new_tokens=1000,
    use_cache=True,
    temperature=0.1,
    min_p=0.1
)

# decode into string
vlm_output = tokenizer.decode(outputs[0], skip_special_tokens=True)
def clean_vlm_output(raw_output: str) -> str:
    # Split on 'assistant' and take everything after it
    if "assistant" in raw_output:
        cleaned = raw_output.split("assistant", 1)[1].strip()
    else:
        cleaned = raw_output.strip()

    return cleaned

cleaned_vlm_output = clean_vlm_output(vlm_output)
print(json.dumps(cleaned_vlm_output, indent=2))

{"'2nd_main_dark_area_present_b4d0': 'occluded',
'2nd_main_dark_area_wing_vein_1952': 'occluded',
'2nd_main_dark_area_wing_vein_828c': 'occluded',
'3rd_main_dark_area_costa_equal_5e00': 'occluded',
'3rd_main_dark_area_much_longer_9b45': 'occluded',
'3rd_main_dark_area_vein_1_990a': 'occluded',
'3rd_main_dark_area_wing_vein_679b': 'occluded',

```

```

'3rd_main_dark_area_wing_vein_bedc': 'occluded',
'abdominal_segments_laterally_projecting_tufts_scales_a455': 'absent',
'abdominal_terga_clothed_yellowish_scales_hindtarsomeres_a1dc':
'occluded', 'all_tarsi_completely_dark_wing_without_c886': 'occluded',
'apex_hindtibia_a_pale_streak_35_f5de': 'present',
'apical_pale_bands_hindtarsomeres_1_4_e889': 'present',
'apices_hindtarsomeres_3_4_dark_most_77ed': 'occluded', 'basal_02f6':
'occluded', 'basal_area_wing_vein_1_proximal_8981': 'occluded',
'basal_pale_band_maxillary_palpus_about_a31a': 'occluded',
'base_costa_1_pale_interruption_3rd_5b6c': 'occluded',
'base_costa_2_pale_interruptions_4fe9': 'occluded',
'base_costa_large_presector_pale_spot_b135': 'occluded',
'base_costa_pale_911e': 'occluded',
'base_costa_two_pale_interruptions_5455': 'occluded',
'base_hindtarsomere_1_dark_pale_fringe_c3b6': 'occluded',
'bases_hindtarsomeres_dark_b530': 'occluded',
'costa_entirely_dark_except_for_a_05f6': 'occluded',
'costa_humeral_pale_spot_no_subapical_4db4': 'occluded',
'erect_head_scales_narrow_rodlike_all_16c1': 'occluded',
'erect_head_scales_narrow_rodlike_all_a14d': 'occluded',
'femora_tibiae_speckled_1830': 'absent',
'fore_middtarsomeres_2_3_pale_apex_4280': 'occluded',
'foretarsomere_1_59_pale_rings_stem_04cc': 'occluded',
'foretarsomere_1_some_speckling_base_costa_0780': 'absent',
'foretarsomere_4_wellmarked_apical_pale_band_0cff': 'occluded',
'foretarsomeres_mainly_pale_narrow_dark_markings_bfbe': 'occluded',
'hind_tarsomere_1_entirely_dark_basally_edcf': 'occluded',
'hindtarsomere_1_broadly_pale_apex_vein_8710': 'occluded',
'hindtarsomere_2_about_apical_04_05_9d0a': 'occluded',
'hindtarsomere_3_entirely_pale_5552': 'occluded',
'hindtarsomere_5_about_apical_05_4_e794': 'occluded',
'hindtarsomere_5_entirely_pale_hindtarsomere_4_035b': 'occluded',
'hindtarsomere_5_mainly_entirely_dark_hindtarsomere_6877': 'occluded',
'hindtarsomeres_1_2_definite_pale_dark_91ac': 'occluded',
'hindtarsomeres_1_4_distinct_apical_pale_c28d': 'present',
'hindtarsomeres_1_4_pale_bands_overlapping_e843': 'present',
'hindtarsomeres_1_5_entirely_dark_81c6': 'occluded',
'hindtarsomeres_2_4_apical_pale_rings_d48e': 'present',
'hindtarsomeres_3_4_all_white_narrowly_2c7a': 'occluded',
'hindtarsomeres_3"

# post process the output
model_json = enforce_json_structure(cleaned_vlm_output,
list(allowed_keys))

import ast

if isinstance(model_json, str):
    model_json = ast.literal_eval(model_json)

```

```
# post process the output
input = decode_feature_json(model_json, basic_key_map)

print(json.dumps(input, indent=2))

{
    "Abdominal segments with laterally projecting tufts of scales on segments II-VII": 0,
        "Hindtarsus with at least last 2 hindtarsomeres entirely pale": 2,
        "Hindtarsomere 5 mainly or entirely dark, hindtarsomere 4 white": 2,
        "Legs speckled, sometimes sparsely": 2,
        "Wing entirely dark or with pale spots confined to costa and vein 1": 2,
        "Wing with at least 1 pale spot on basal 0.5 of costa": 2,
        "Maxillary palpus with apex dark": 2,
        "Maxillary palpus with 4 pale bands": 2,
        "Wing with pale interruption in 3rd main dark area (preapical dark spot) of vein 1, sometimes fused with preceding pale area": 2,
        "Wing with 2 pale spots on upper branch of vein 5": 2,
        "Wing almost entirely dark, costa without pale spots": 2,
        "Hindtarsomeres 1 to 5 entirely dark": 2,
        "Hindtarsomeres 1 and 2 with definite pale and dark rings in addition to apical pale bands": 2,
        "Hindtarsomeres 3 and 4 all white or narrowly dark basally, 5 all dark or at least basal 0.5 dark": 2,
        "Moderate-sized species; abdominal scale-tufts short and dark; 0.5 or more of hindtarsomere 1 pale": 2,
        "Pale fringe spot present opposite lower branch of vein 5": 2,
        "Hindtarsomere 5 and about apical 0.5 of 4 pale": 2,
        "Very small species (wing length 2.5-2.8 mm); wing with upper branch of vein 2 largely pale": 2,
        "Legs speckled": 2,
        "Hindtarsomeres 3 to 5 entirely pale": 2,
        "Maxillary palpus with 3 pale bands, usually with some speckling; vein 1 of wing with 2 pale spots in 2nd main dark area (median dark spot)": 2,
        "Midtarsomeres 2 to 4 entirely dark; vein 1 of wing dark at base, basal 0.5 of stem of vein 4 with small pale areas": 2,
        "Hindtarsomere 1 broadly pale at apex; vein 1 of wing with 2 pale spots in 2nd main dark area": 2,
        "Foretarsomere 1 with 5-9 pale rings; stem of vein 4 of wing largely pale": 2,
        "Fore- and midtarsomeres 2 and 3 pale at apex; wing with fringe spot opposite vein 6": 2,
        "Maxillary palpus very shaggy and unbanded or with 1-4 irregular narrow pale bands": 2,
        "Maxillary palpus with 1-4 pale bands; apex of hindtibia broadly or narrowly pale": 2,
        "Hindtarsomere 3 entirely pale": 2,
        "Base of hindtarsomere 1 dark; pale fringe spot present opposite
```

lower branch of wing vein 5": 2,
"Hind tarsomere 1 entirely dark basally or at most with a very narrow band of pale scales not as broad as the width of the tarsomere": 2,
"Apex of hindtibia with a pale streak 3-5 times as long as broad; apical pale band on hindtarsomere 2 0.13-0.4 length of tarsomere": 1,
"3rd main dark area on wing vein 1 without a pale interruption; foretarsomeres 1 to 3 usually without distinct apical pale bands": 2,
"Femora and tibiae speckled": 0,
"Maxillary palpus shaggy; costa and vein 1 of wing without usual main dark areas": 2,
"Maxillary palpus with 3 pale bands": 2,
"Maxillary palpus with apical 2 pale bands very broad, speckling on palpus segment 3; 2nd main dark area on wing vein 1 with 2 pale interruptions": 2,
"3rd main dark area of wing vein 1 with a pale interruption, sometimes fused with preceding pale spot; scaling on abdomen very scanty, confined to tergum VIII or rarely VII": 2,
"Maxillary palpus speckled": 2,
"Foretarsomere 1 with some speckling; base of costa with 2 pale spots; stem of wing vein 2 entirely pale": 0,
"All tarsi completely dark; wing without pale fringe spots posterior to vein 3": 2,
"3rd main dark area of vein 1 with a pale interruption, sometimes fused with preceding pale area": 2,
"Hindtarsomere 2 with about apical 0.4 to 0.5 white and the rest dark": 2,
"Hindtarsomeres 2 to 4 with apical pale rings and otherwise dark except for 1 to 2 pale spots; no pale fringe spot opposite wing vein 6": 1,
"Foretarsomeres mainly pale with narrow dark markings": 2,
"Scales on abdominal tergum VIII dense and distributed over whole tergum, sometimes with a few scales on lateral borders of tergum VII": 2,
"Wings entirely dark or unicolorous": 2,
"Maxillary palpus with 2 well-marked pale bands; hindfemur and hindtibia narrowly pale at apex": 2,
"Large species, wing length 4 mm or more": 2,
"Very pale brown species with glossy scutum; semi-arid regions only": 2,
"Maxillary palpus with 2 to 3 pale bands, pale at apex (sometimes indistinct)": 2,
"Erect head scales narrow, rod-like, all scales yellowish throughout; semi-arid regions only": 2,
"Pale and dark areas on wing poorly contrasted; semi-arid regions only": 2,
"Pale areas on wing very narrow, subcostal pale spot present on costa only; cave-dwelling": 2,
"Maxillary palpus with 3 pale bands, dark at apex": 2,

"Semi-arid regions; pale brown species with poorly contrasting light and dark areas on wing": 2,
"Erect head scales narrow, rod-like, all scales yellowish throughout": 2,
"Maxillary palpus shaggy to near tip": 2,
"Maxillary palpus entirely dark; hindtarsomeres 3 and 4 dark or narrowly pale at apices": 2,
"Maxillary palpus with apex dark, sometimes only narrowly so": 2,
"Stem of wing vein 4 largely pale, upper branch of vein 5 with 2 pale spots or largely pale, fringe spots present opposite vein 4 and upper branch of vein 5": 2,
"Costa entirely dark except for a few indistinct pale scales subapically; maxillary palpus with a broad apical pale band and otherwise dark except for a narrow basal pale band": 2,
"Maxillary palpus with 3 pale bands, subapical band broad and about equal in length to apical band": 2,
"Wing, apart from costa, generally very pale, basal 0.5 of stems of veins 2 and 4 entirely pale": 2,
"Pale fringe spots present opposite all veins from wing apex to vein 5, stem of vein 5 broadly dark near base": 2,
"Wing vein 5 entirely dark except for a single pale spot on the upper branch": 2,
"Hindtarsomeres 1 to 4 with distinct apical pale bands; scutum clothed with very narrow scales": 1,
"Median scutal scales yellowish or bronze, white elsewhere": 2,
"Maxillary palpus entirely dark or without distinct pale bands": 2,
"Small, pale brown species, pale patches on wing indistinct, basal 0.25-0.5 of costa entirely dark; head scales narrow and yellowish": 2,
"Costa with humeral pale spot, no subapical (preapical) pale spot on costa and vein 1": 2,
"Wing with pale fringe spots opposite all veins except vein 6": 2,
"Wing generally pale, contrast between pale and dark areas, apart from costa and vein 1, poorly defined": 2,
"2nd main dark area of wing vein 1 with 2 pale interruptions": 2,
"Pale bands on maxillary palpus very narrow, at apices of segments 2 to 4 and not overlapping the joints; upper branch of wing vein 5 with a single pale spot": 2,
"Wing, apart from costa and vein 1, predominantly dark, no pale spots on basal 0.25 of costa": 2,
"Wing vein 6 with proximal pale spot": 2,
"Basal 0.2 of wing vein 1 either dark or with a proximal pale patch not extending to base": 2,
"Wings scantily scaled, all wing scales very narrow": 2,
"Basal pale band of maxillary palpus about equal to or slightly shorter than median band, broadly overlapping base of 3rd segment": 2,
"Base of costa pale": 2,
"3rd main dark area of wing vein 1 with a pale interruption": 2,
"Abdominal terga clothed with yellowish scales; hindtarsomeres 1 to 4 with broad apical pale bands": 2,

"Second main dark area of wing vein 1 with 2 pale interruptions": 2,
"Pale bands on maxillary palpus broad, basal band overlapping base of 3rd segment": 2,
"Pale fringe spots on wing present opposite veins posterior to vein 3, sometimes including vein 6; femora and tibiae not speckled": 2,
"Stem of wing vein 5 pale, at and adjacent to the fork": 2,
"Wing length 4 mm or less; decumbent scutal scales not extending onto scutellum": 2,
"2nd and 3rd main dark areas present on vein 1": 2,
"Hindtarsomere 5 entirely pale, hindtarsomere 4 with broad apical and basal pale bands": 2,
"Upper branch of wing vein 5 with 1 pale spot, sometimes a vestigial 2nd pale spot": 2,
"Pale fringe spot present opposite wing vein 6; foretarsomeres 1 to 4 with conspicuous basal and apical pale bands": 2,
"Subapical pale band on maxillary palpus very narrow, confined to apex of 3rd segment": 2,
"Wing with base of costa with 2 pale interruptions": 2,
"Hindtarsomeres either all dark or with pale bands on tarsomeres 1 and 2 only": 2,
"Scutal fossae and lateral areas of scutum above wing root (supraalar area) with scattered or abundant broadish scales": 2,
"Subapical pale band on maxillary palpus about equal to or slightly shorter than apical band": 2,
"Apical pale bands on hindtarsomeres 1 to 4 very broad, at least twice the apical width of the tarsomeres": 1,
"Wing vein 3 largely dark or broadly dark at either end; scutal scales very narrow and golden": 2,
"Scutal scales as in (A, B, C or D)": 2,
"Maxillary palpus with only apical pale band": 2,
"Subapical pale band on maxillary palpus broad, about equal to or longer than apical dark band": 2,
"Bases of hindtarsomeres dark": 2,
"2nd main dark area on wing vein 1 with 2 pale interruptions; bases of hindtarsomeres 4 and 5 broadly or narrowly pale": 2,
"Base of costa with 1 pale interruption, 3rd main dark area on costa and vein 1 much broader than subcostal pale spot": 2,
"Pale fringe spot present opposite wing vein 6": 2,
"Apices of hindtarsomeres 3 and 4 dark or at most with a few pale scales": 2,
"Base of costa with 2 pale interruptions": 2,
"Wing vein 6 either with pale fringe spot or with pale scales at apex of vein": 2,
"Scutal scales fairly broad, extending over whole scutum and onto scutellum": 2,
"Very small species, wing length 2.8 mm or less": 2,
"Hindtarsomeres entirely dark; preaccessory dark spot on wing vein 1 usually absent": 2,
"3rd main dark area much longer than subapical pale spot": 2,

```

    "Scutal scales broadish and white, only slightly less dense on
posterior 0.33 of scutum than anteriorly, and extending onto
scutellum": 2,
    "Moderate-sized species, wing length more than 3.2 mm": 2,
    "Foretarsomere 4 with well-marked apical pale band; wing with fringe
spot opposite vein 6": 2,
    "3rd main dark area of costa equal to or shorter than subapical pale
spot": 2,
    "Wing with fork of vein 5 pale": 2,
    "Base of costa with two pale interruptions": 2,
    "Small species, wing length about 2.4-3.3 mm": 2,
    "Maxillary palpus with only apex pale": 2,
    "Base of costa with large (presector) pale spot, base of vein 1
pale": 2,
    "Lower branch of wing vein 2 and upper branch of vein 4 with
distinct pale spots": 2,
    "Subapical pale spot on costa and wing vein 1 about as long as
apical dark spot, fringe spots present opposite veins 3, lower branch
of 4 and both branches of 5": 2,
    "Hindtarsomeres 1 to 4 with pale bands overlapping the joints, at
least hindtarsomere 5 pale basally": 1,
    "Preaccessory dark spot on wing vein 1 about twice as long as pale
spot on either side of it": 2,
    "Basal area of wing vein 1 proximal to 1st main dark area, pale with
a broad dark spot": 2,
    "Subapical pale band on maxillary palpus longer than or equal to
apical dark band AND 3rd main dark area of costa and vein 1 equal to
or shorter than subapical pale spot": 2,
    "Moderate-sized species, wing length more than 3.3 mm": 2,
    "Tip of wing vein 6 dark with no fringe spot": 2
}

# process through the BN
pred, _ = predict_species(BN_model, input, verbose=True)
print(pred)

```

```

==== Bayesian Network Prediction ====
Predicted species: brumpti
Probability: 0.37490575037560037

```

```

Full distribution:
ardensis: 0.0000
argenteolobatus: 0.0001
aruni: 0.0000
austenii: 0.0000
azaniae: 0.0000
azevedoi: 0.0000
barberellus: 0.0000
berghei: 0.0000

```

bervoetsi: 0.0000
brohieri: 0.0000
brucei: 0.0000
brumpti: 0.3749
brunnipes: 0.0000
buxtoni: 0.0000
caliginosus: 0.0000
cameroni: 0.0000
carnevalei: 0.0000
caroni: 0.0000
carteri: 0.0000
christyi: 0.0000
cinctus: 0.0000
cinereus: 0.0000
concolor: 0.0000
coustani: 0.0000
cristipalpis: 0.0000
crypticus: 0.0000
culicifacies: 0.0000
cydippis: 0.0000
discalicus: 0.0000
daudi: 0.0000
deemingi: 0.0000
demeilloni: 0.0000
demeilloni (Berg River form): 0.0000
distinctus: 0.0000
domicolus: 0.0000
dthali: 0.0000
dureni: 0.0000
erepens: 0.0000
faini: 0.0000
flavicosta: 0.0000
fontinalis: 0.0000
freetownensis: 0.0000
funestus: 0.0000
fuscivenosus: 0.0000
gabonensis: 0.0000
gambiae: 0.0000
garnhami: 0.0000
gibbinsi: 0.0000
hamoni: 0.0000
hancocki: 0.0000
hargreavesi: 0.0000
harperi: 0.0000
hervyi: 0.0000
hughi: 0.0000
implexus: 0.0000
jebudensis: 0.0000
keniensis: 0.0000

kingi: 0.2500
kosiensis: 0.0000
letabensis: 0.0000
listeri: 0.0000
lloreti: 0.0000
longipalpis: 0.0000
lounibosi: 0.0000
lovettae: 0.0000
machardyti: 0.0000
maculipalpis: 0.0000
maliensis: 0.0000
marshallii: 0.0000
millecampsi: 0.0000
mortiauxi: 0.0000
moucheti: 0.0000
mousinholi: 0.0000
multicinctus: 0.0000
multicolor: 0.0000
murphyi: 0.0001
namibiensis: 0.0000
nan: 0.0000
natalensis: 0.0000
nili: 0.0000
nili (Congo form): 0.0000
njombiensis: 0.0000
obscurus: 0.0000
okuensis: 0.0000
ovengensis: 0.0000
paludis: 0.0000
parensis: 0.0000
pharoensis: 0.0000
pretoriensis: 0.0000
rageaui: 0.0000
rhodesiensis: 0.0000
rivulorum: 0.0000
rodhaini: 0.0000
ruarinus: 0.0000
rufipes: 0.0000
salbaii: 0.0000
schwetzi: 0.0000
seretsei: 0.0000
sergentii: 0.0000
seydeli: 0.0000
smithii: 0.0000
somalicus: 0.0000
squamosus: 0.0000
stephensi: 0.0000
swahilicus: 0.0000
symesi: 0.3749

```

tchekedii: 0.0000
tenebrosus: 0.0000
turkhudi: 0.0000
vanhoofi: 0.0000
vernus: 0.0000
vinckeii: 0.0000
walravensi: 0.0000
wellcomei: 0.0000
wilsoni: 0.0000
ziemannii: 0.0000
brumpti

# what did the original predict?
test = enforce_json_structure(dataset[im]["caption"], basic_key_map)
test = decode_feature_json(test, basic_key_map)

def convert_for_bn(decoded, reverse_map):
    out = {}
    for full_text, val in decoded.items():
        if val in [0, 1] and full_text in reverse_map:
            short_key = reverse_map[full_text]
            out[short_key] = val
    return out

test = convert_for_bn(test, basic_key_map)

# process through the BN
pred2, _, _ = predict_species(BN_model, test, verbose=True)

==== Bayesian Network Prediction ====
Predicted species: ardensis
Probability: 0.008620689655172412

Full distribution:
ardensis: 0.0086
argenteolobatus: 0.0086
aruni: 0.0086
austenii: 0.0086
azaniae: 0.0086
azevedoi: 0.0086
barberellus: 0.0086
berghei: 0.0086
bervoetsi: 0.0086
brohieri: 0.0086
brucei: 0.0086
brumpti: 0.0086
brunnipes: 0.0086
buxtoni: 0.0086
caliginosus: 0.0086

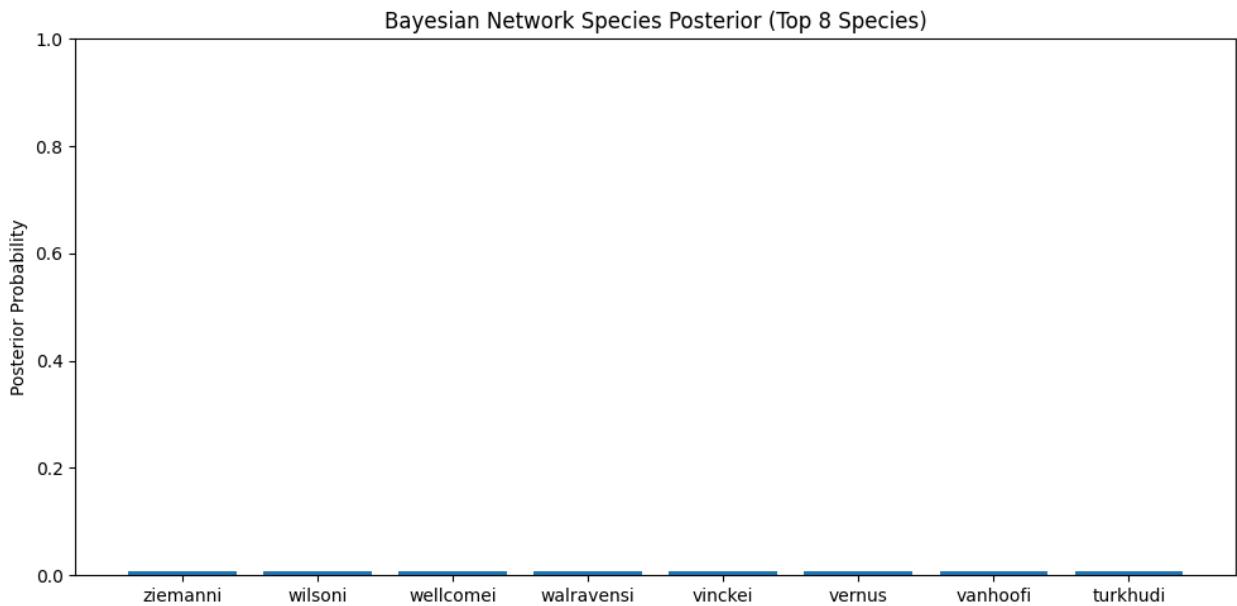
```

cameroni: 0.0086
carnevalei: 0.0086
caroni: 0.0086
carteri: 0.0086
christyi: 0.0086
cinctus: 0.0086
cinereus: 0.0086
concolor: 0.0086
coustani: 0.0086
cristipalpis: 0.0086
crypticus: 0.0086
culicifacies: 0.0086
cydippis: 0.0086
dancalicus: 0.0086
daudi: 0.0086
deemingi: 0.0086
demeilloni: 0.0086
demeilloni (Berg River form): 0.0086
distinctus: 0.0086
domicolus: 0.0086
dthali: 0.0086
dureni: 0.0086
erepens: 0.0086
faini: 0.0086
flavicosta: 0.0086
fontinalis: 0.0086
freetownensis: 0.0086
funestus: 0.0086
fuscivenosus: 0.0086
gabonensis: 0.0086
gambiae: 0.0086
garnhami: 0.0086
gibbinsi: 0.0086
hamoni: 0.0086
hancocki: 0.0086
hargreavesi: 0.0086
harperi: 0.0086
hervyi: 0.0086
hughi: 0.0086
implexus: 0.0086
jebudensis: 0.0086
keniensis: 0.0086
kingi: 0.0086
kosiensis: 0.0086
letabensis: 0.0086
listeri: 0.0086
lloretii: 0.0086
longipalpis: 0.0086
lounibosi: 0.0086

lovettae: 0.0086
machardyi: 0.0086
maculipalpis: 0.0086
maliensis: 0.0086
marshallii: 0.0086
millecampsi: 0.0086
mortiauxi: 0.0086
moucheti: 0.0086
mousinhoi: 0.0086
multicinctus: 0.0086
multicolor: 0.0086
murphyi: 0.0086
namibiensis: 0.0086
nan: 0.0086
natalensis: 0.0086
nili: 0.0086
nili (Congo form): 0.0086
njombiensis: 0.0086
obscurus: 0.0086
okuensis: 0.0086
ovengensis: 0.0086
paludis: 0.0086
parensis: 0.0086
pharoensis: 0.0086
pretoriensis: 0.0086
rageaui: 0.0086
rhodesiensis: 0.0086
rivulorum: 0.0086
rodhaini: 0.0086
ruarinus: 0.0086
rufipes: 0.0086
salbaii: 0.0086
schwetzi: 0.0086
seretsei: 0.0086
sergentii: 0.0086
seydeli: 0.0086
smithii: 0.0086
somalicus: 0.0086
squamosus: 0.0086
stephensi: 0.0086
swahilicus: 0.0086
symesi: 0.0086
tchekedii: 0.0086
tenebrosus: 0.0086
turkhudi: 0.0086
vanhoofi: 0.0086
vernus: 0.0086
vinckeii: 0.0086
walravensi: 0.0086
wellcomei: 0.0086

```
wilsoni: 0.0086
ziemanni: 0.0086

# plot the distribution
pred2, _, posterior = predict_species(BN_model, test, verbose=False)
species, probs = posterior
plot_posterior(species, probs, 8)
```



```
# renormalization for fun

def restrict_and_renormalize(query_result, allowed_subset):
    """
    Given a pgmpy query_result for variable 'Species',
    filter to a subset of species and renormalize probabilities.

    Returns:
        best_name      - the MAP species inside allowed_subset
        best_prob      - its renormalized probability
        filtered_names - list of species kept
        filtered_probs - renormalized probabilities
    """
    # Extract original posterior
    species_names = query_result.state_names["Species"]
    probs = query_result.values # array

    # Filter
    filtered_names = []
    filtered_probs = []

    for name, prob in zip(species_names, probs):
```

```

        if name in allowed_subset:
            filtered_names.append(name)
            filtered_probs.append(prob)

    filtered_probs = np.array(filtered_probs)

    # Handle edge case: none found
    if filtered_probs.sum() == 0:
        # Return uniform over subset
        uniform_prob = 1.0 / len(filtered_names)
        filtered_probs = np.array([uniform_prob] * len(filtered_names))
    else:
        # Renormalize
        filtered_probs /= filtered_probs.sum()

    # Pick best
    max_idx = filtered_probs.argmax()
    best_name = filtered_names[max_idx]
    best_prob = filtered_probs[max_idx]

    return best_name, best_prob, filtered_names, filtered_probs

inference = VariableElimination(BN_model)
allowed_subset = ["funestus", "gambiae", "tenebrosus", "pharoensis",
"coustani"]

query_result = inference.query(
    variables=["Species"],
    evidence=input
)

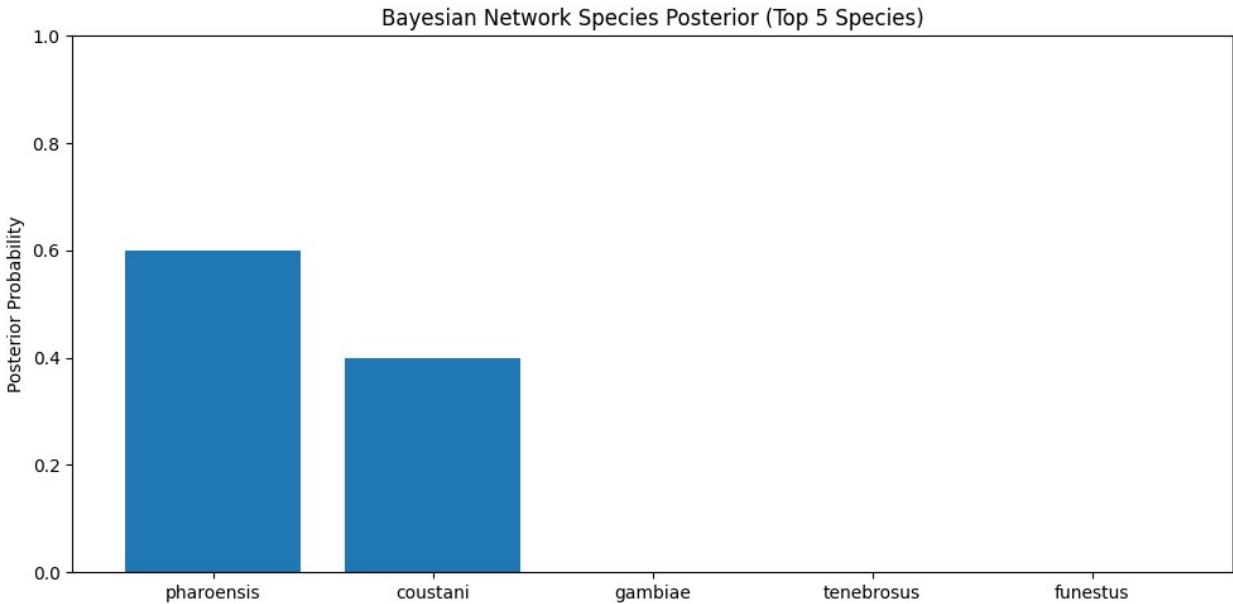
best_name, best_prob, names, probs =
restrict_and_renormalize(query_result, allowed_subset)

print("Most likely species:", best_name)
print("Probability:", best_prob)
print("Distribution:")
for n, p in zip(names, probs):
    print(f" {n}: {p:.3f}")

Most likely species: pharoensis
Probability: 0.5999655093210257
Distribution:
coustani: 0.400
funestus: 0.000
gambiae: 0.000
pharoensis: 0.600
tenebrosus: 0.000

plot_posterior(names, probs, 5)

```



Bleu and Rouge Scores

note that it doesn't make sense to use these scores because they measure semantic similarity. This is just a demonstration. The actual metric for structured output should be a simple accuracy score for correct key output

```
%%capture
!pip install evaluate
!pip install nltk
!pip install rouge_score
import evaluate

print(json.dumps(cleaned_vlm_output, indent=2))
print(json.dumps(dataset[im]["caption"]))

{"2nd_main_dark_area_present_b4d0": "occluded",
 "2nd_main_dark_area_wing_vein_1952": "occluded",
 "2nd_main_dark_area_wing_vein_828c": "occluded",
 "3rd_main_dark_area_costa_equal_5e00": "occluded",
 "3rd_main_dark_area_much_longer_9b45": "occluded",
 "3rd_main_dark_area_vein_1_990a": "occluded",
 "3rd_main_dark_area_wing_vein_679b": "occluded",
 "3rd_main_dark_area_wing_vein_bedc": "occluded",
 "abdominal_segments_laterally_projecting_tufts_scales_a455": "absent",
 "abdominal_terga_clothed_yellowish_scales_hindtarsomeres_a1dc": "occluded",
 "all_tarsi_completely_dark_wing_without_c886": "occluded",
 "apex Hindtibia_a_pale_streak_35_f5de": "present",
 "apical_pale_bands_hindtarsomeres_1_4_e889": "present",
 "apices_hindtarsomeres_3_4_dark_most_77ed": "occluded", "basal_02f6": "occluded"}
```

```
'occluded', 'basal_area_wing_vein_1_proximal_8981': 'occluded',
'basal_pale_band_maxillary_palpus_about_a31a': 'occluded',
'base_costa_1_pale_interruption_3rd_5b6c': 'occluded',
'base_costa_2_pale_interruptions_4fe9': 'occluded',
'base_costa_large_presector_pale_spot_b135': 'occluded',
'base_costa_pale_911e': 'occluded',
'base_costa_two_pale_interruptions_5455': 'occluded',
'base_hindtarsomere_1_dark_pale_fringe_c3b6': 'occluded',
'bases_hindtarsomeres_dark_b530': 'occluded',
'costa_entirely_dark_except_for_a_05f6': 'occluded',
'costa_humeral_pale_spot_no_subapical_4db4': 'occluded',
'erect_head_scales_narrow_rodlike_all_16c1': 'occluded',
'erect_head_scales_narrow_rodlike_all_a14d': 'occluded',
'femora_tibiae_speckled_1830': 'absent',
'fore_middatarsomeres_2_3_pale_apex_4280': 'occluded',
'foretarsomere_1_59_pale_rings_stem_04cc': 'occluded',
'foretarsomere_1_some_speckling_base_costa_0780': 'absent',
'foretarsomere_4_wellmarked_apical_pale_band_0cff': 'occluded',
'foretarsomeres_mainly_pale_narrow_dark_markings_bfbe': 'occluded',
'hind_tarsomere_1_entirely_dark_basally_edcf': 'occluded',
'hindtarsomere_1_broadly_pale_apex_vein_8710': 'occluded',
'hindtarsomere_2_about_apical_04_05_9d0a': 'occluded',
'hindtarsomere_3_entirely_pale_5552': 'occluded',
'hindtarsomere_5_about_apical_05_4_e794': 'occluded',
'hindtarsomere_5_entirely_pale_hindtarsomere_4_035b': 'occluded',
'hindtarsomere_5_mainly_entirely_dark_hindtarsomere_6877': 'occluded',
'hindtarsomeres_1_2_definite_pale_dark_91ac': 'occluded',
'hindtarsomeres_1_4_distinct_apical_pale_c28d': 'present',
'hindtarsomeres_1_4_pale_bands_overlapping_e843': 'present',
'hindtarsomeres_1_5_entirely_dark_81c6': 'occluded',
'hindtarsomeres_2_4_apical_pale_rings_d48e': 'present',
'hindtarsomeres_3_4_all_white_narrowly_2c7a': 'occluded',
'hindtarsomeres_3"
{"2nd_3rd_main_dark_areas_present_b4d0": "occluded",
"2nd_main_dark_area_wing_vein_1952": "occluded",
"2nd_main_dark_area_wing_vein_828c": "occluded",
"3rd_main_dark_area_costa_equal_5e00": "occluded",
"3rd_main_dark_area_much_longer_8b45": "occluded",
"3rd_main_dark_area_vein_1_990a": "occluded",
"3rd_main_dark_area_wing_vein_679b": "occluded",
"3rd_main_dark_area_wing_vein_bedc": "occluded",
"3rd_main_dark_area_wing_vein_e54a": "occluded",
"abdominal_segments_laterally_projecting_tufts_scales_a455": "present",
"abdominal_terga_clothed_yellowish_scales_hindtarsomeres_a1dc": "occluded", "all_tarsi_completely_dark_wing_without_c886": "absent",
"apex Hindtibia_a_pale_streak_35_f5de": "occluded",
"apical_pale_bands_hindtarsomeres_1_4_e889": "present",
"apices_hindtarsomeres_3_4_dark_most_77ed": "absent",
```

"basal_02_wing_vein_1_either_edee": "occluded",
"basal_area_wing_vein_1_proximal_8981": "occluded",
"basal_pale_band_maxillary_palpus_about_a31a": "occluded",
"base_costa_1_pale_interruption_3rd_5b6c": "present",
"base_costa_2_pale_interruptions_4fe9": "present",
"base_costa_large_presector_pale_spot_b135": "present",
"base_costa_pale_911e": "present",
"base_costa_two_pale_interruptions_5455": "present",
"base_hindtarsomere_1_dark_pale_fringe_c3b6": "occluded",
"bases_hindtarsomeres_dark_b530": "absent",
"costa_entirely_dark_except_for_a_05f6": "absent",
"costa_humeral_pale_spot_no_subapical_4db4": "present",
"erect_head_scales_narrow_rodlike_all_16c1": "occluded",
"erect_head_scales_narrow_rodlike_all_a14d": "occluded",
"femora_tibiae_speckled_1830": "occluded",
"fore_middtarsomeres_2_3_pale_apex_4280": "occluded",
"foretarsomere_1_59_pale_rings_stem_04cc": "occluded",
"foretarsomere_1_some_speckling_base_costa_0780": "occluded",
"foretarsomere_4_wellmarked_apical_pale_band_0cff": "occluded",
"foretarsomeres_mainly_pale_narrow_dark_markings_bfbe": "occluded",
"hind_tarsomere_1_entirely_dark_basally_edcf": "absent",
"hindtarsomere_1_broadly_pale_apex_vein_8710": "occluded",
"hindtarsomere_2_about_apical_04_05_9d0a": "present",
"hindtarsomere_3_entirely_pale_5552": "absent",
"hindtarsomere_5_about_apical_05_4_e794": "present",
"hindtarsomere_5_entirely_pale_hindtarsomere_4_035b": "absent",
"hindtarsomere_5_mainly_entirely_dark_hindtarsomere_6877": "absent",
"hindtarsomeres_1_2_definite_pale_dark_91ac": "present",
"hindtarsomeres_1_4_distinct_apical_pale_c28d": "present",
"hindtarsomeres_1_4_pale_bands_overlapping_e843": "present",
"hindtarsomeres_1_5_entirely_dark_81c6": "absent",
"hindtarsomeres_2_4_apical_pale_rings_d48e": "present",
"hindtarsomeres_3_4_all_white_narrowly_2c7a": "absent",
"hindtarsomeres_3_5_entirely_pale_0256": "absent",
"hindtarsomeres_either_all_dark_pale_bands_7e48": "absent",
"hindtarsomeres_entirely_dark_preaccessory_dark_spot_ffcd": "absent",
"hindtarsus_least_last_2_hindtarsomeres_entirely_13b3": "occluded",
"large_species_wing_length_4_mm_c3fb": "occluded",
"legs_speckled_d1c4": "occluded",
"legs_speckled_sometimes_sparsely_5829": "occluded",
"lower_branch_wing_vein_2_upper_f86b": "occluded",
"maxillary_palpus_14_pale_bands_apex_0dad": "occluded",
"maxillary_palpus_2_3_pale_bands_7736": "occluded",
"maxillary_palpus_2_wellmarked_pale_bands_25a0": "occluded",
"maxillary_palpus_3_pale_bands_dark_3c64": "occluded",
"maxillary_palpus_3_pale_bands_fa8e": "occluded",
"maxillary_palpus_3_pale_bands_subapical_ba70": "occluded",
"maxillary_palpus_3_pale_bands_usually_a6ba": "occluded",
"maxillary_palpus_4_pale_bands_80b3": "occluded",

"maxillary_palpus_apex_dark_3f74": "occluded",
"maxillary_palpus_apex_dark_sometimes_only_9cac": "occluded",
"maxillary_palpus_apical_2_pale_bands_36b7": "occluded",
"maxillary_palpus_entirely_dark_hindtarsomeres_3_00e0": "occluded",
"maxillary_palpus_entirely_dark_without_distinct_2919": "occluded",
"maxillary_palpus_only_apex_pale_4509": "occluded",
"maxillary_palpus_only_apical_pale_band_1645": "occluded",
"maxillary_palpus_shaggy_costa_vein_1_108c": "occluded",
"maxillary_palpus_shaggy_near_tip_923f": "occluded",
"maxillary_palpus_speckled_d59c": "occluded",
"maxillary_palpus_very_shaggy_unbanded_14_64e2": "occluded",
"median_scutal_scales_yellowish_bronze_white_1e48": "occluded",
"midtarsomeres_2_4_entirely_dark_vein_9e10": "occluded",
"moderatesized_species_abdominal_scaletufts_short_dark_6a66":
"occluded", "moderatesized_species_wing_length_more_than_18ea":
"occluded", "moderatesized_species_wing_length_more_than_77ff":
"occluded", "pale_areas_wing_very_narrow_subcostal_bd00": "present",
"pale_bands_maxillary_palpus_broad_basal_cb18": "occluded",
"pale_bands_maxillary_palpus_very_narrow_b1cd": "occluded",
"pale_dark_areas_wing_poorly_contrasted_f6d1": "present",
"pale_fringe_spot_present_opposite_lower_afce": "occluded",
"pale_fringe_spot_present_opposite_wing_624e": "occluded",
"pale_fringe_spot_present_opposite_wing_8ec2": "occluded",
"pale_fringe_spots_present_opposite_all_0586": "occluded",
"pale_fringe_spots_wing_present_opposite_7852": "occluded",
"preaccessory_dark_spot_wing_vein_1_91d1": "occluded",
"scales_abdominal_tergum_viii_dense_distributed_05f0": "occluded",
"scutal_fossae_lateral_areas_scutum_above_7c02": "occluded",
"scutal_scales_as_a_b_c_f1b3": "occluded",
"scutal_scales_broadish_white_only_slightly_260a": "occluded",
"scutal_scales_fairly_broad_extending_over_b9f3": "occluded",
"second_main_dark_area_wing_vein_31a0": "occluded",
"semiarid_regions_pale_brown_species_poorly_d91b": "occluded",
"small_pale_brown_species_pale_patches_00ea": "occluded",
"small_species_wing_length_about_2433_0108": "occluded",
"stem_wing_vein_4_largely_pale_092e": "occluded",
"stem_wing_vein_5_pale_adjacent_4882": "occluded",
"subapical_pale_band_maxillary_palpus_about_4270": "occluded",
"subapical_pale_band_maxillary_palpus_broad_f967": "occluded",
"subapical_pale_band_maxillary_palpus_longer_eb4a": "occluded",
"subapical_pale_band_maxillary_palpus_very_2ef5": "occluded",
"subapical_pale_spot_costa_wing_vein_c38f": "occluded",
"tip_wing_vein_6_dark_no_589f": "occluded",
"upper_branch_wing_vein_5_1_c9a3": "occluded",
"very_pale_brown_species_glossy_scutum_13c7": "occluded",
"very_small_species_wing_length_2528_486e": "occluded",
"very_small_species_wing_length_28_dff4": "occluded",
"wing_2_pale_spots_upper_branch_e992": "occluded",
"wing_almost_entirely_dark_costa_without_aeaa": "absent",

```

"wing_apart_from_costa_generally_very_236e": "present",
"wing_apart_from_costa_vein_1_64c4": "occluded",
"wing_base_costa_2_pale_interruptions_6765": "present",
"wing_entirely_dark_pale_spots_confined_a50b": "absent",
"wing_fork_vein_5_pale_f718": "occluded",
"wing_generally_pale_contrast_between_pale_7919": "present",
"wing_least_1_pale_spot_basal_43dd": "present",
"wing_length_4_mm_less_decumbent_f06f": "occluded",
"wing_pale_fringe_spots_opposite_all_f40b": "occluded",
"wing_pale_interruption_3rd_main_dark_dfc8": "occluded",
"wing_vein_3_largely_dark_broadly_8993": "occluded",
"wing_vein_5_entirely_dark_except_c75f": "occluded",
"wing_vein_6_either_pale_fringe_0d24": "occluded",
"wing_vein_6_proximal_pale_spot_e127": "occluded",
"wings_entirely_dark_unicolorous_fda3": "absent",
"wings_scantily_scaled_all_wing_scales_53f1": "absent"}
```

#Compute Metrics

Load the metrics

```

bleu = evaluate.load("bleu")
rouge = evaluate.load("rouge")
```

set up the refs

```

ref1 = dataset[im]["caption"]
```

set up the predictions

Example data (can fill it with the dichotomous key stuff later...)

```

predictions = [
    json.dumps(input, indent=2),
]
```

```

references = [
    json.dumps(ref1, indent=2),
]
```

bleu

```

bleu_result = bleu.compute(
    predictions=predictions,
    references=references
)
```

rouge

```

rouge_result = rouge.compute(
    predictions=predictions,
    references=[r[0] for r in references]
)
```

```

print("BLEU:", json.dumps(bleu_result, indent=4))
print("ROUGE:", json.dumps(rouge_result, indent=4))

{"model_id": "18d87c01f0d4430caa13c754e7e59ff9", "version_major": 2, "version_minor": 0}

{"model_id": "093820d338ef44b7a5d9a41bfa55bc24", "version_major": 2, "version_minor": 0}

{"model_id": "e9b9854b47394dc2b3dc09a5a442dc56", "version_major": 2, "version_minor": 0}

{"model_id": "687a2a4fa95146b3b3d6dc09985e84", "version_major": 2, "version_minor": 0}

BLEU: {
    "bleu": 0.0,
    "precisions": [
        0.4615085536547434,
        0.10462854920264489,
        0.0031128404669260703,
        0.0
    ],
    "brevity_penalty": 1.0,
    "length_ratio": 1.0226640159045726,
    "translation_length": 2572,
    "reference_length": 2515
}
ROUGE: {
    "rouge1": 0.0,
    "rouge2": 0.0,
    "rougeL": 0.0,
    "rougeLsum": 0.0
}

```

Accuracy of Inference Output Structure

```

print(json.dumps(cleaned_vlm_output))
print(json.dumps(dataset[im]["caption"]))

{"'2nd_main_dark_area_present_b4d0': 'occluded',
 '2nd_main_dark_area_wing_vein_1952': 'occluded',
 '2nd_main_dark_area_wing_vein_828c': 'occluded',
 '3rd_main_dark_area_costa_equal_5e00': 'occluded',
 '3rd_main_dark_area_much_longer_9b45': 'occluded',
 '3rd_main_dark_area_vein_1_990a': 'occluded',
 '3rd_main_dark_area_wing_vein_679b': 'occluded',
 '3rd_main_dark_area_wing_vein_bedc': 'occluded',
 'abdominal_segments_laterally_projecting_tufts_scales_a455': 'absent',

```

```
'abdominal_terga_clothed_yellowish_scales_hindtarsomeres_a1dc':  
'occluded', 'all_tarsi_completely_dark_wing_without_c886': 'occluded',  
'apex_hindtibia_a_pale_streak_35_f5de': 'present',  
'apical_pale_bands_hindtarsomeres_1_4_e889': 'present',  
'apices_hindtarsomeres_3_4_dark_most_77ed': 'occluded', 'basal_02f6':  
'occluded', 'basal_area_wing_vein_1_proximal_8981': 'occluded',  
'basal_pale_band_maxillary_palpus_about_a31a': 'occluded',  
'base_costa_1_pale_interruption_3rd_5b6c': 'occluded',  
'base_costa_2_pale_interruptions_4fe9': 'occluded',  
'base_costa_large_presector_pale_spot_b135': 'occluded',  
'base_costa_pale_911e': 'occluded',  
'base_costa_two_pale_interruptions_5455': 'occluded',  
'base_hindtarsomere_1_dark_pale_fringe_c3b6': 'occluded',  
'bases_hindtarsomeres_dark_b530': 'occluded',  
'costa_entirely_dark_except_for_a_05f6': 'occluded',  
'costa_humeral_pale_spot_no_subapical_4db4': 'occluded',  
'erect_head_scales_narrow_rodlike_all_16c1': 'occluded',  
'erect_head_scales_narrow_rodlike_all_a14d': 'occluded',  
'femora_tibiae_speckled_1830': 'absent',  
'fore_middtarsomeres_2_3_pale_apex_4280': 'occluded',  
'foretarsomere_1_59_pale_rings_stem_04cc': 'occluded',  
'foretarsomere_1_some_speckling_base_costa_0780': 'absent',  
'foretarsomere_4_wellmarked_apical_pale_band_0cff': 'occluded',  
'foretarsomeres_mainly_pale_narrow_dark_markings_bfbe': 'occluded',  
'hind_tarsomere_1_entirely_dark_basally_edcf': 'occluded',  
'hindtarsomere_1_broadly_pale_apex_vein_8710': 'occluded',  
'hindtarsomere_2_about_apical_04_05_9d0a': 'occluded',  
'hindtarsomere_3_entirely_pale_5552': 'occluded',  
'hindtarsomere_5_about_apical_05_4_e794': 'occluded',  
'hindtarsomere_5_entirely_pale_hindtarsomere_4_035b': 'occluded',  
'hindtarsomere_5_mainly_entirely_dark_hindtarsomere_6877': 'occluded',  
'hindtarsomeres_1_2_definite_pale_dark_91ac': 'occluded',  
'hindtarsomeres_1_4_distinct_apical_pale_c28d': 'present',  
'hindtarsomeres_1_4_pale_bands_overlapping_e843': 'present',  
'hindtarsomeres_1_5_entirely_dark_81c6': 'occluded',  
'hindtarsomeres_2_4_apical_pale_rings_d48e': 'present',  
'hindtarsomeres_3_4_all_white_narrowly_2c7a': 'occluded',  
'hindtarsomeres_3"  
{"2nd_3rd_main_dark_areas_present_b4d0": "occluded",  
"2nd_main_dark_area_wing_vein_1952": "occluded",  
"2nd_main_dark_area_wing_vein_828c": "occluded",  
"3rd_main_dark_area_costa_equal_5e00": "occluded",  
"3rd_main_dark_area_much_longer_8b45": "occluded",  
"3rd_main_dark_area_vein_1_990a": "occluded",  
"3rd_main_dark_area_wing_vein_679b": "occluded",  
"3rd_main_dark_area_wing_vein_bedc": "occluded",  
"3rd_main_dark_area_wing_vein_e54a": "occluded",  
"abdominal_segments_laterally_projecting_tufts_scales_a455":  
"present",
```

"abdominal_terga_clothed_yellowish_scales_hindtarsomeres_aldc":
"occluded", "all_tarsi_completely_dark_wing_without_c886": "absent",
"apex_hindtibia_a_pale_streak_35_f5de": "occluded",
"apical_pale_bands_hindtarsomeres_1_4_e889": "present",
"apices_hindtarsomeres_3_4_dark_most_77ed": "absent",
"basal_02_wing_vein_1_either_edee": "occluded",
"basal_area_wing_vein_1_proximal_8981": "occluded",
"basal_pale_band_maxillary_palpus_about_a31a": "occluded",
"base_costa_1_pale_interruption_3rd_5b6c": "present",
"base_costa_2_pale_interruptions_4fe9": "present",
"base_costa_large_presector_pale_spot_b135": "present",
"base_costa_pale_911e": "present",
"base_costa_two_pale_interruptions_5455": "present",
"base_hindtarsomere_1_dark_pale_fringe_c3b6": "occluded",
"bases_hindtarsomeres_dark_b530": "absent",
"costa_entirely_dark_except_for_a_05f6": "absent",
"costa_humeral_pale_spot_no_subapical_4db4": "present",
"erect_head_scales_narrow_rodlike_all_16c1": "occluded",
"erect_head_scales_narrow_rodlike_all_a14d": "occluded",
"femora_tibiae_speckled_1830": "occluded",
"fore_middtarsomeres_2_3_pale_apex_4280": "occluded",
"foretarsomere_1_59_pale_rings_stem_04cc": "occluded",
"foretarsomere_1_some_speckling_base_costa_0780": "occluded",
"foretarsomere_4_wellmarked_apical_pale_band_0cff": "occluded",
"foretarsomeres_mainly_pale_narrow_dark_markings_bfbe": "occluded",
"hind_tarsomere_1_entirely_dark_basally_edcf": "absent",
"hindtarsomere_1_broadly_pale_apex_vein_8710": "occluded",
"hindtarsomere_2_about_apical_04_05_9d0a": "present",
"hindtarsomere_3_entirely_pale_5552": "absent",
"hindtarsomere_5_about_apical_05_4_e794": "present",
"hindtarsomere_5_entirely_pale_hindtarsomere_4_035b": "absent",
"hindtarsomere_5_mainly_entirely_dark_hindtarsomere_6877": "absent",
"hindtarsomeres_1_2_definite_pale_dark_91ac": "present",
"hindtarsomeres_1_4_distinct_apical_pale_c28d": "present",
"hindtarsomeres_1_4_pale_bands_overlapping_e843": "present",
"hindtarsomeres_1_5_entirely_dark_81c6": "absent",
"hindtarsomeres_2_4_apical_pale_rings_d48e": "present",
"hindtarsomeres_3_4_all_white_narrowly_2c7a": "absent",
"hindtarsomeres_3_5_entirely_pale_0256": "absent",
"hindtarsomeres_either_all_dark_pale_bands_7e48": "absent",
"hindtarsomeres_entirely_dark_preaccessory_dark_spot_ffcd": "absent",
"hindtarsus_least_last_2_hindtarsomeres_entirely_13b3": "occluded",
"large_species_wing_length_4_mm_c3fb": "occluded",
"legs_speckled_d1c4": "occluded",
"legs_speckled_sometimes_sparsely_5829": "occluded",
"lower_branch_wing_vein_2_upper_f86b": "occluded",
"maxillary_palpus_14_pale_bands_apex_0dad": "occluded",
"maxillary_palpus_2_3_pale_bands_7736": "occluded",
"maxillary_palpus_2_wellmarked_pale_bands_25a0": "occluded",

"maxillary_palpus_3_pale_bands_dark_3c64": "occluded",
"maxillary_palpus_3_pale_bands_fa8e": "occluded",
"maxillary_palpus_3_pale_bands_subapical_ba70": "occluded",
"maxillary_palpus_3_pale_bands_usually_a6ba": "occluded",
"maxillary_palpus_4_pale_bands_80b3": "occluded",
"maxillary_palpus_apex_dark_3f74": "occluded",
"maxillary_palpus_apex_dark_sometimes_only_9cac": "occluded",
"maxillary_palpus_apical_2_pale_bands_36b7": "occluded",
"maxillary_palpus_entirely_dark_hindtarsomeres_3_00e0": "occluded",
"maxillary_palpus_entirely_dark_without_distinct_2919": "occluded",
"maxillary_palpus_only_apex_pale_4509": "occluded",
"maxillary_palpus_only_apical_pale_band_1645": "occluded",
"maxillary_palpus_shaggy_costa_vein_1_108c": "occluded",
"maxillary_palpus_shaggy_near_tip_923f": "occluded",
"maxillary_palpus_speckled_d59c": "occluded",
"maxillary_palpus_very_shaggy_unbanded_14_64e2": "occluded",
"median_scutal_scales_yellowish_bronze_white_1e48": "occluded",
"midtarsomeres_2_4_entirely_dark_vein_9e10": "occluded",
"moderatesized_species_abdominal_scaletufts_short_dark_6a66":
"occluded", "moderatesized_species_wing_length_more_than_18ea":
"occluded", "moderatesized_species_wing_length_more_than_77ff":
"occluded", "pale_areas_wing_very_narrow_subcostal_bd00": "present",
"pale_bands_maxillary_palpus_broad_basal_cb18": "occluded",
"pale_bands_maxillary_palpus_very_narrow_b1cd": "occluded",
"pale_dark_areas_wing_poorly_contrasted_f6d1": "present",
"pale_fringe_spot_present_opposite_lower_afce": "occluded",
"pale_fringe_spot_present_opposite_wing_624e": "occluded",
"pale_fringe_spot_present_opposite_wing_8ec2": "occluded",
"pale_fringe_spots_present_opposite_all_0586": "occluded",
"pale_fringe_spots_wing_present_opposite_7852": "occluded",
"preaccessory_dark_spot_wing_vein_1_91d1": "occluded",
"scales_abdominal_tergum_viii_dense_distributed_05f0": "occluded",
"scutal_fossae_lateral_areas_scutum_above_7c02": "occluded",
"scutal_scales_as_a_b_c_f1b3": "occluded",
"scutal_scales_broadish_white_only_slightly_260a": "occluded",
"scutal_scales_fairly_broad_extending_over_b9f3": "occluded",
"second_main_dark_area_wing_vein_31a0": "occluded",
"semiarid_regions_pale_brown_species_poorly_d91b": "occluded",
"small_pale_brown_species_pale_patches_00ea": "occluded",
"small_species_wing_length_about_2433_0108": "occluded",
"stem_wing_vein_4_largely_pale_092e": "occluded",
"stem_wing_vein_5_pale_adjacent_4882": "occluded",
"subapical_pale_band_maxillary_palpus_about_4270": "occluded",
"subapical_pale_band_maxillary_palpus_broad_f967": "occluded",
"subapical_pale_band_maxillary_palpus_longer_eb4a": "occluded",
"subapical_pale_band_maxillary_palpus_very_2ef5": "occluded",
"subapical_pale_spot_costa_wing_vein_c38f": "occluded",
"tip_wing_vein_6_dark_no_589f": "occluded",
"upper_branch_wing_vein_5_1_c9a3": "occluded",

```

"very_pale_brown_species_glossy_scutum_13c7": "occluded",
"very_small_species_wing_length_2528_486e": "occluded",
"very_small_species_wing_length_28_dff4": "occluded",
"wing_2_pale_spots_upper_branch_e992": "occluded",
"wing_almost_entirely_dark_costa_without_aeaa": "absent",
"wing_apart_from_costa_generally_very_236e": "present",
"wing_apart_from_costa_vein_1_64c4": "occluded",
"wing_base_costa_2_pale_interruptions_6765": "present",
"wing_entirely_dark_pale_spots_confined_a50b": "absent",
"wing_fork_vein_5_pale_f718": "occluded",
"wing_generally_pale_contrast_between_pale_7919": "present",
"wing_least_1_pale_spot_basal_43dd": "present",
"wing_length_4_mm_less_decumbent_f06f": "occluded",
"wing_pale_fringe_spots_opposite_all_f40b": "occluded",
"wing_pale INTERRUPTION_3rd_main_dark_dfc8": "occluded",
"wing_vein_3_largely_dark_broadly_8993": "occluded",
"wing_vein_5_entirely_dark_except_c75f": "occluded",
"wing_vein_6_either_pale_fringe_0d24": "occluded",
"wing_vein_6_proximal_pale_spot_e127": "occluded",
"wings_entirely_dark_unicolorous_fda3": "absent",
"wings_scantily_scaled_all_wing_scales_53f1": "absent"}
```

compare model_json vs dataset[im]["caption"] keys

```

pred_keys = set(model_json.keys())
actual_keys = set(dataset[im]["caption"].keys())

jaccard = len(actual_keys & pred_keys) / len(actual_keys | pred_keys)
print('Jaccard overlap: ', jaccard)

Jaccard overlap:  1.0
```