Data Methodology and Sources

December 2024

Data Set Qualifications

Protected Areas

Access to accurate, comprehensive, and up-to-date geospatial data on protected areas (PAs) in China, particularly in the Tibet region, is highly limited due to far-reaching government restrictions on geographic data in the Chinese territory. A comprehensive dataset is difficult to obtain and would likely require a request from governmental agencies, such as the Ministry of Environment.

To create a more complete dataset, different sources providing partial data coverage were integrated. The World Database on Protected Areas (WDPA) (UNEP-WCMC, 2024) provides the most up-to-date dataset for China; however, it primarily contains internationally designated protected areas (PAs) and is therefore incomplete, with the WDPA website stating that in China, “2,960 protected areas are not publicly available”. We sourced earlier data of WDPA from March 2018, which is no longer accessible from the WDPA website, from Gillespie et al. (2019). While covering a substantial amount of additional national nature reserves, this dataset is somewhat outdated and does not include provincial and county-level PAs. We also acquired one further PA data object each from OpenStreetMap (OSM 2024) and the International Centre for Integrated Mountain Development (ICIMOD 2015). Although this provides a more comprehensive picture, the resulting data set stemming from these diverse sources has to be treated with a high level of caution, as its accuracy and degree of spatial coverage remains uncertain.

Additionally, due to China's regulations on coordinate obfuscation in public map services, there is likely an offset of 50-500 meters in the data of nationally designated PAs (Bizyayeva, 2024). This offset further compromises definitive statements about the proximity of dam sites to these PAs. Data of internationally designated PAs (such as Ramsar or World Heritage Sites) are probably not affected by this issue.

Land Cover

We used the ESRI 10m Annual Land Cover dataset (Karra et al., 2021), produced by Impact Observatory, of the year 2023 for our land cover analysis. This dataset is derived from ESA Sentinel-2 imagery at 10m resolution and includes predictions for nine land cover classes. The dataset was generated using a deep learning model trained on billions of human-labeled pixels curated by the National Geographic Society. The validation of the dataset yielded an average accuracy of over 75%.

Population Counts

To assess the presence of human populations within the area of influence of the proposed dams, we used the Oak Ridge National Laboratory's LandScan Global dataset for 2022, which is considered an industry standard for global population distribution (Sims et al. 2022). This

dataset provides population estimates at a 30 arc-second (~1 km) resolution. Using a multivariable dasymetric modeling approach, the underlying algorithm spatially disaggregates recent census data within administrative boundaries into pixel-level estimates by leveraging machine learning, geographic data, and satellite imagery.

A study by Ma et al. (2021) compared LandScan population estimates with census-based data from municipalities in the Tibet region. The results demonstrated that LandScan has reasonable predictive performance in Tibet, with a coefficient of determination (R^2) of 0.91 and a root mean squared error (RMSE) of 151,927. These findings support the qualification of this dataset for our analyses in the Tibetan region, but it should be noted that further uncertainties remain in the population estimates, particularly in small and rural study areas and areas with nomadic populations. In these areas, human populations tend to be geographically more dispersed with fewer or less pronounced anthropogenic markers, making accurate spatial quantification more difficult.

Settlement points

Settlement point data was obtained from OpenStreetMap (OSM) via the Humanitarian Data Exchange (HDX, 2024) catalog. It includes a range of settlement types such as cities, towns, villages and hamlets with precise location information. Although not necessarily complete, this dataset provides important information on human settlements that further support and contextualize the analysis of population distribution in the impact areas of the proposed or existing dams.

Cultural Sites

To identify Tibetan cultural heritage sites within the assumed impact areas of the dams, we used vector geodata from the Treasury of Lives web map, an encyclopedic resource of Tibetan biographies (Treasury of Lives, 2024). This geodata encompasses culturally significant locations related to biographies featured on the website, such as monasteries, temples, and sacred landscape elements. Sites within the 50km buffer zones of the case studies were individually verified and relocated.

Methodology - Zonal Statistics

Areas of Impact definitions

Web Map areas of impact:

Our methodology attempts to provide a structured approach for defining the spatial influence of dams, by integrating both distance from the dam and proximity to river networks.

For each dam, we generated a circular buffer proportional to the dam's size with respect to its hydroelectric power capacity. The buffer represents the potential upstream and downstream influence extending from the dam. After creating the circular buffers, we identified the river systems within the area of influence using the HydroRIVERS database. We generated a linear buffer along the rivers within the circular buffer, with distances set at 2 km for larger dams and 1 km for smaller dams. This ensures the removal of areas outside the dam's hydrological reach in the analysis.

Larger dams tend to have more extensive river systems within their buffer zones, leading to a more circular final area. In contrast, smaller dams result in smaller circular buffers, capturing fewer rivers and resulting in more rectangular-shaped areas of influence.

Tunnel-based dams:

For dams located at the end of tunnels, our methodology places the dam's center of influence halfway along the tunnel, often in an area of higher elevation, and therefore excluded from the HydroRIVERS derived river buffer. For example, the center of Motuo Dam's area of influence falls outside the defined boundary. Though this may initially appear counterintuitive, this methodology enables us to include both the entrance and exit to the tunnel in our analysis.

Whilst helpful in providing initial estimates, these buffers do not fully account for the complexity of the social and environmental processes that are taking place. Overestimation can occur where the buffer zones extend into areas where the dam has no influence, such as non-hydrologically connected tributaries. Conversely, underestimation can occur where there are indirect or long-range impacts, for example a reduction in river flow causing irrigation issues downstream. Additionally, our approach assumes a uniform impact within the buffer, even though the actual impact might vary in intensity depending on social, environmental, topographic, and hydrological conditions in the region. More detailed analyses may be conducted in future iterations of this project.

Below is the table we used for this analysis:

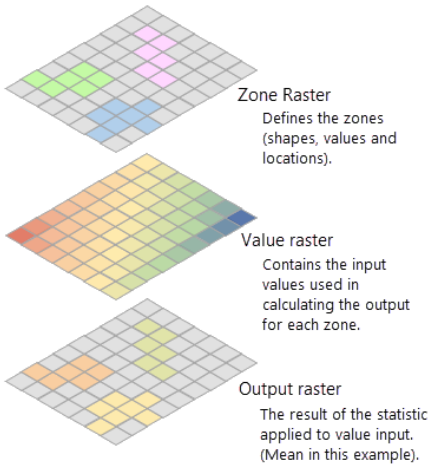
Hydroelectric Capacity (MW)	Dam Classification (# Dams)	River Buffer (km)	Upstream/Downstream Buffer (km)
0–10	*Small (8)*	*1 km*	*2.5 km*
10–100	*Medium (27)*	*1 km*	*5 km*
100–500	*Large (58)*	*2 km*	*10 km*
500+	*Mega (94)*	*2 km*	*25 km*
Unknown	*Null (6)*	*1.5 km*	*15 km*

Case study dams:

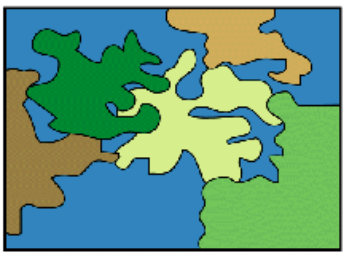
To characterize the societal and ecological conditions surrounding the sites of 4 particularly risky and contentious dams ("case study dams": Motuo, Gangtuo, Yangqu, and Lianghekou) we created a buffer zone of 50 km around each dam site. While somewhat arbitrary, this allows for an initial assessment of the surrounding land cover and land use characteristics as well as the presence of human population using zonal statistics.

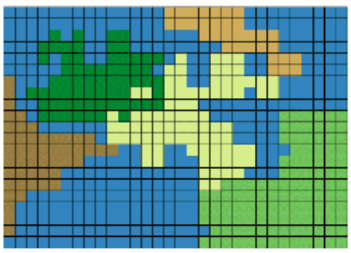
Zonal statistics

Zonal statistics is a spatial analysis technique that calculates statistical values for a raster dataset within a specified geographic area (zone). It requires two data set inputs: a value raster containing the variable of interest (here population or land cover), and a zone raster (or "rasterized" vector) defining the spatial boundaries of each zone (here dam buffer zones). The workflow process involves overlaying the zone raster on the value raster, extracting the values within each zone, and then calculating statistics based on these extracted values (e.g. sum, mean, median, frequency) (Garrard, 2016, pp. 258–263).



Rasterization of vector data

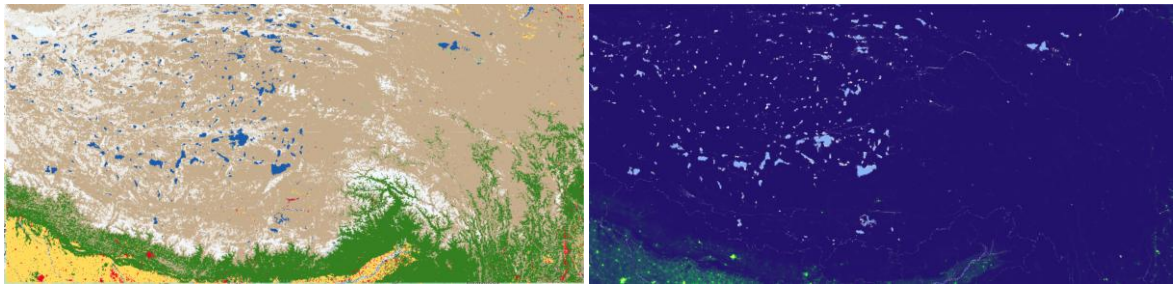

Polygon features


Raster polygon features

https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/zonal-statistics.htm

Zone rasters - rasterized dam areas of impact

Value rasters - land cover and population data



We conducted a zonal statistical analysis to assess both population counts and land cover distribution within the dam buffer zones as defined above (assumed impact areas). For the population analysis, we used the Landscan Global data product (Sims et al. 2023), which has a spatial resolution of 30 arc seconds (~1 km). For the land cover analysis we used the ESRI 10m Annual Land Cover dataset (Karra et al., 2021) for the year 2023 with a resolution of 10m.

We iteratively rasterized each dam buffer feature to obtain the zone raster, aligned with the respective population and land cover datasets, before calculating the statistics (e.g., population sums and land cover type area sizes). We used various open-source Python libraries, including GDAL, NumPy, Pandas, and GeoPandas for the creation of this workflow.

We used the ESRI:102025 (Asia North Albers Equal Area Conic) coordinate system to calculate land cover area statistics. This projection is designed for northern and central Asia and preserves area measures. Its unit of measure is meters, which allows for straightforward conversion of the pixel-based figures into area measures. For this, the land cover raster data was reprojected from its original geographic coordinate system (EPSG:4326) to ESRI:102025 using nearest neighbor resampling.

Population proximity analysis

For the case study maps, we generated plots of cumulative population sums as a function of distance from dam sites. To do this, we first created a proximity raster using GDAL, aligned to the spatial characteristics of the LandScan raster, with values representing Euclidean distances from dam sites. This proximity raster was then overlaid with the LandScan population data to determine the population values for each distance. The pairs of values (population, distance to dam site) were extracted and compiled into a data frame that allowed the calculation and subsequent visualisation of cumulative population numbers at increasing distances from the dams.

References

Bizyayeva, A. (2024). Every map of China is wrong. *Medium*. https://medium.com/@anastasia.bizyayeva/every-map-of-china-is-wrong-bc2bce145db2

Esri. (2010). *World Major Rivers* [Dataset]. https://www.arcgis.com/home/item.html?id=44e8358cf83a4b43bc863646cd695945

Garrard, C. (2016). *Geoprocessing with Python*. Manning Publications Co. LLC.

Gillespie, T. W., Madson, A., Cusack, C. F., & Xue, Y. (2019). Changes in NDVI and human population in protected areas on the Tibetan Plateau. *Arctic, Antarctic, and Alpine Research*, *51*(1), 428–439. https://doi.org/10.1080/15230430.2019.1650541

Humanitarian Data Exchange (HDX). (2017). *China Waterways (OpenStreetMap Export)*. Retrieved from https://data.humdata.org and originally sourced from OpenStreetMap contributors, https://www.openstreetmap.org.

Humanitarian Data Exchange (HDX). (2024). China Populated Places (OpenStreetMap Export). Retrieved from https://data.humdata.org and originally sourced from OpenStreetMap contributors, https://www.openstreetmap.org.

ICIMOD. (2015). *Boundary of Kailash Sacred Landscape* [Dataset]. ICIMOD. https://doi.org/10.26066/RDS.22037

Karra, K., Kontgis, C., Statman-Weil, Z., Mazzariello, J. C., Mathis, M., & Brumby, S. P. (2021). Global land use / land cover with Sentinel 2 and deep learning. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 4704–4707. https://doi.org/10.1109/IGARSS47720.2021.9553499

Lehner, B., & Grill, G. (2013). Global river hydrography and network routing: Baseline data and new approaches to study the world's large river systems. *Hydrological Processes*, *27*(15), 2171–2186. https://doi.org/10.1002/hyp.9740

Ma, X., Yang, Z., Wang, J., & Han, F. (2022). Mapping population on Tibetan Plateau by fusing VIIRS data and nighttime Tencent location-based services data. *Ecological Indicators*, *139*, 108893. https://doi.org/10.1016/j.ecolind.2022.108893

Sims, K., Reith, A., Bright, E., Kaufman, J., Pyle, J., Epting, J., Gonzales, J., Adams, D., Powell, E., Urban, M., & Rose, A. (2023). *LandScan Global 2022* [Data set]. Oak Ridge National Laboratory. https://doi.org/10.48690/1529167

Treasury of Lives. (2024). *Places* [Dataset]. https://treasuryoflives.org/aboutmaps

UNEP-WCMC. (2024). *Protected Area Profile for China from the World Database on Protected Areas* [Dataset]. www.protectedplanet.net

OpenStreetMap contributors. (2024). *OSM data of protected areas.* [Dataset] https://www.openstreetmap.org. Retrieved using the Python library OSMnx (tags: 'boundary': 'protected_area').

Zonal Statistics Analyses code

```
import os
import numpy as np
import pandas as pd
import geopandas as gpd
from osgeo import gdal
import glob


# Load dam buffer shapefile and create lists for iteration
dam_buff_gdf = gpd.read_file("tibet_dams_buffer.shp")
dam_ids_l = dam_buff_gdf['DamID'].astype(int).tolist()
names_l = dam_buff_gdf['Name'].tolist()

# ========================= # Land cover analysis # ============================== #
raster_dir_l = [os.path.join("lc_data", filename) for filename in os.listdir("lc_data") if filename.endswith(".tif")] #
store land cover file locations in list (raster files have been reprojected to ESRI:102025 )

# Initialize dictionary to store land cover area results
lc_lists_dict = {
    'water_km2_l': [], 'trees_km2_l': [], 'flooded_vegetation_km2_l': [],
    'crops_km2_l': [], 'built_area_km2_l': [], 'bare_ground_km2_l': [],
    'snow/ice_km2_l': [], 'clouds_km2_l': [], 'rangeland_km2_l': [],
    'total_buffer_area_km2': []
}

# Iterate over each dam buffer
for i, dam_id in enumerate(dam_ids_l):
    print(f"Land Cover Analysis for: {names_l[i]}")

    # Subset the buffer and align CRS with raster files
    subset_dam_buff_gdf = dam_buff_gdf[dam_buff_gdf['DamID'] == dam_id].to_crs('ESRI:102025')
    subset_dam_buff_gdf.to_file('temp_subset_dam_buff.shp')  # Export temporary shapefile for the buffer

    # Get buffer bounding box coordinates
    bbox = subset_dam_buff_gdf.total_bounds  # [minx, miny, maxx, maxy]
    lc_temp_arr = np.array([])  # Temporary array to store land cover data for the buffer
    images_count = 0  # Count of overlapping land cover tiles

    # Loop through land cover raster tiles
    for lc_tile_file in raster_dir_l:
        ds = gdal.Open(lc_tile_file)
        lc_arr = ds.ReadAsArray().astype(np.uint8)
        gt = ds.GetGeoTransform()

        # Define raster extent for overlap check
        raster_bbox = [gt[0], gt[3] + ds.RasterYSize * gt[5], gt[0] + ds.RasterXSize * gt[1], gt[3]]
        pixel_area = abs(gt[1]) * abs(gt[5])

        # Check if raster tile overlaps with buffer
```

```
        if not (bbox[2] < raster_bbox[0] or bbox[0] > raster_bbox[2] or bbox[3] < raster_bbox[1] or bbox[1] > raster_bbox[3]):
            images_count += 1

            # Rasterize buffer to match land cover raster's spatial properties
            gdal.Rasterize(
                "tibet_dams_buff_rasterized.tif", "temp_subset_dam_buff.shp", format='GTiff',
                outputType=gdal.GDT_Int32, creationOptions=["COMPRESS=DEFLATE"], noData=0,
                xRes=gt[1], yRes=gt[5], outputBounds=[raster_bbox[0], raster_bbox[1], raster_bbox[2], raster_bbox[3]],
                attribute='DamID'
            )

            # Read rasterized buffer as an array and mask by the dam ID
            buff_arr = gdal.Open("tibet_dams_buff_rasterized.tif").ReadAsArray()
            masked_lc_arr = lc_arr[buff_arr == dam_id]

            # Concatenate masked array for area calculations
            lc_temp_arr = np.concatenate((lc_temp_arr, masked_lc_arr))
            print("lc_array_length:", len(lc_temp_arr))

    print(f"{images_count} tiles overlap with buffer")

    # Calculate and store area for each land cover type
    for lc_id in lc_names_map.keys():
        area_km2 = np.count_nonzero(lc_temp_arr == lc_id) * pixel_area / 1_000_000  # km²
        lc_lists_dict[lc_lists_dict_map[lc_id]].append(area_km2)  # Append to respective land cover list

ds = None # close data sets

# ============================ # Population Analysis # ============================
# Initialize population count list
pop_sum_l = []

# Open Landscan population raster and load data
ds = gdal.Open("landscan_pop_tibet_2022.tif")
pop_arr = np.array(ds.GetRasterBand(1).ReadAsArray()).astype(np.float64)
pop_arr[pop_arr == -2147483647.0] = 0  # Replace no data value with 0 for Landscan

# Retrieve raster metadata for buffer rasterization
gt = ds.GetGeoTransform()
lx, uy = gt[0], gt[3]
rx, ly = gt[0] + ds.RasterXSize * gt[1], gt[3] + ds.RasterYSize * gt[5]

# Loop through each dam buffer zone
for i, dam_id in enumerate(dam_ids_l):
    print(f"Population Analysis for: {names_l[i]}")

    # Subset the buffer feature for the specific dam
    subset_dam_buff_gdf = dam_buff_gdf[dam_buff_gdf['DamID'] == dam_id]
    subset_dam_buff_gdf.to_file('temp_subset_dam_buff.shp') # create temporary shapefile

    # Rasterize dam buffer feature
```

```
    gdal.Rasterize(
        "tibet_dams_buff_rasterized.tif",
        "temp_subset_dam_buff.shp",
        format='GTiff',
        outputType=gdal.GDT_Int32,
        creationOptions=["COMPRESS=DEFLATE"],
        noData=0,
        xRes=gt[1], yRes=gt[5],
        outputBounds=[lx, ly, rx, uy],
        attribute='DamID'
    )

    # Read buffer raster and mask population np array
    ds_buffer = gdal.Open("tibet_dams_buff_rasterized.tif")
    buff_arr = np.array(ds_buffer.GetRasterBand(1).ReadAsArray())
    masked_pop_arr = pop_arr[buff_arr == dam_id]

    # Calculate and store statistics in list
    pop_sum = np.sum(masked_pop_arr)
    pop_sum_l.append(round(pop_sum))
    ds_buffer = None  # Close the buffer raster

ds = None  # Close the population raster data set
os.remove("tibet_dams_buff_rasterized.tif") # delete temporary data
for i in glob.glob('temp_subset_dam_buff.*'): # delete all subfiles
    os.unlink(i)

# ========================== # Zonal Statistics Data Frame # ==========================
zonal_stats_df = pd.DataFrame({
    "dam_id": dam_ids_l,
    "dam_name": names_l,
    "water_km2": lc_lists_dict["water_km2_l"],
    "trees_km2": lc_lists_dict["trees_km2_l"],
    "flooded_vegetation_km2": lc_lists_dict["flooded_vegetation_km2_l"],
    "crops_km2": lc_lists_dict["crops_km2_l"],
    "built_area_km2": lc_lists_dict["built_area_km2_l"],
    "bare_ground_km2": lc_lists_dict["bare_ground_km2_l"],
    "snow_ice_km2": lc_lists_dict["snow/ice_km2_l"],
    "clouds_km2": lc_lists_dict["clouds_km2_l"],
    "rangeland_km2": lc_lists_dict["rangeland_km2_l"],
    "population_sum": pop_sum_l,
})

zonal_stats_df.to_excel('zonal_stats.xlsx', sheet_name=zonal_stats, index=False)  # export df to excel
```