A EQUIVARIANCE AND SYMMETRY

Equivariant neural networks are designed to explicitly incorporate symmetries that are present in the underlying data. Symmetries, often derived from first principles or domain knowledge, such as rotational or translational invariance, allow the network to process inputs in a way that is consistent with these transformations. This is particularly important when the ground truth functions respect such symmetries, as the incorporation of these properties can significantly enhance model performance and generalization.

Group A group of symmetries or simply *group* is a set G together with a binary operation $\circ\colon G \times G \to G$ called *composition* satisfying three properties: 1) *identity*: There is an element $1 \in G$ such that $1 \circ g = g \circ 1 = g$ for all $g \in G$; 2) *associativity*: $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$ for all $g_1, g_2, g_3 \in G$; 3) *inverses* if $g \in G$, then there is an element $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = 1$.

Examples of groups include the dihedral groups D_4 (symmetries of a square) and D_8 (symmetries of an octagon), as well as the orthogonal group $O(2)$, which represents all rotations and reflections in 2D space. Both D_4 and D_8 are discrete subgroups of $O(2)$.

Representation A group representation defines how a group action transforms elements of a vector space by mapping group elements to linear transformations on that space. More specifically, a group representation of a group G on a vector space V is is a homomorphism: $\rho\colon G \to \mathrm{GL}(X)$, where $\mathrm{GL}(X)$ is the group of invertible linear transformations on V. This means for any $g_1, g_2 \in G$, ρ is a linear transformation (often represented by a matrix) such that the group operation in G is preserved:

$$\rho(g_1 g_2) = \rho(g_1)\rho(g_2) \tag{3}$$

Equivariance Formally, a neural network is said to be equivariant to a group of transformations G if applying a transformation from the group to the input results in a corresponding transformation to the output. Mathematically, for a function $f\colon X \to Y$ to be **G-equivariant**, the following condition must hold:

$$f(\rho_{\text{in}}(g)(x)) = \rho_{\text{out}}(g) f(x) \tag{4}$$

for all $x \in X$ and $g \in G$, where $\rho_{\text{in}}\colon G \to \mathrm{GL}(X)$ and $\rho_{\text{out}}\colon G \to \mathrm{GL}(Y)$ are input and output representations (Bronstein et al., 2021). Invariance is a special case of equivariance where the output does not change under the group action. This occurs when the output representation $\rho_{\text{out}}(g)$ is trivial. Figure 8 visualize how the equivariant and invariant networks work.

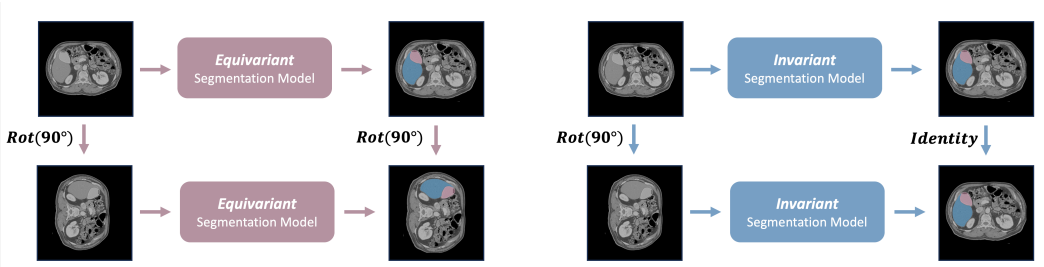

Figure 8: An equivariant model (left) ensures that its output transforms in a specific, predictable way under a group of transformations applied to the input, preserving the structure of the transformation (e.g., rotating the input results in a correspondingly rotated output). In contrast, an invariant model (right) produces an output that remains unchanged regardless of any transformations applied to the input from the same group.

Equivariance via weight-sharing One of the primary approaches to incorporating symmetry into neural networks is through weight sharing (Satorras et al., 2021; Cohen et al., 2018; Wang et al.). This approach enforces equivariance by constraining the network's architecture so that the weights are shared across different group elements. For example, in G-convolutions (Cohen & Welling, 2016), the same set of weights is shared across the transformed versions of the input, ensuring that the network's predictions remain consistent under those transformations. In a layer of G-steerable

CNNs (Weiler & Cesa, 2019), a set of equivariant kernel bases is precomputed based on the input and output representations, and the convolution kernel used is a linear combination of this equivariant kernel basis set, where the coefficients are trainable. Similar approaches can also be used to develop equivariant graph neural networks (Geiger & Smidt, 2022). These architectures directly modify the network's layers to be equivariant, ensuring that each layer processes symmetries in a way that is aligned with the desired group. While powerful, this approach imposes architectural constraints, which may limit the flexibility of the network and prevent leveraging large pretrained models.

Equivariance via canonicalization An alternative to weight sharing is incorporating symmetry through canonicalization (Kaba et al., 2022; Mondal et al., 2023), where, instead of modifying the network's architecture to handle symmetries, the input data is transformed into a canonical form. In this approach, a separate canonicalization network, which is itself equivariant, preprocesses the input, transforming it into a standard, or canonical, representation. This canonicalized input is then passed to a standard prediction network that does not need to be aware of the symmetries. If the corresponding inverse transformation is applied to the output of the prediction network, the entire model becomes equivariant; otherwise, the model remains invariant. This method has several advantages. First, it does not require altering the architecture of the prediction network, allowing for the use of large pre-trained models without modification. Second, by ensuring that the input data is in a canonical form, the prediction network only needs to learn the mapping from the canonical input to the output, without needing to learn all transformed samples. This can lead to improved performance and robustness, especially in scenarios where the prediction task does not naturally align with the symmetry group or where architectural constraints might hinder performance. Thus, in our work, we leverage canonicalization to achieve equivariance in the segmentation task. By transforming the input into a canonical form using a simple equivariant canonicalization network, we ensure that our prediction network remains unconstrained and can fully utilize its capacity for learning without the need for architectural modifications. This approach offers the benefits of symmetry-aware processing while maintaining the flexibility and power of unconstrained neural network architectures.

B Detailed Dataset Description

Image Data Collection and Preprocessing For model development and evaluation, we collected 1,437 CT scans from 7 public datasets. A detailed summary of the datasets is provided in Table 5. In total, 24 organs are labled in the assembled datasets, with a strong focus on segmentation targets in the abdominal region. The organ class distribution across the datasets is shown in Fig 9. To standardize quality and reduce domain gaps, we applied a preprocessing pipeline to all datasets. Specifically, we mapped the Hounsfield unit range [-180, 240] to [0, 1], clipping values outside this range. To address dimension mismatches between datasets, masks, and images, all scans and masks were resized to 1024×1024. The 3D scan volumes were sliced along the axial plane to generate 2D images and corresponding masks. To ensure labeling quality, organ segments with fewer than 1,000 pixels in 3D volumes or fewer than 100 pixels in 2D slices were excluded. The finalized dataset consisted of 101,217 images, with 91,344 (90.25%) used for training and validation, and 9,873 (9.75%) reserved for testing.

Table 5: Overview of the datasets used in this study.

Dataset	# Training scans	# Testing scans	Annotated organs[1]
AbdomenCT-1K	722	—	Liv, Kid, Spl, Pan
MSD[2]	157	—	Lun, Spl
WORD	100	20	Liv, Spl, LKid, RKid, Sto, Gal, Eso, Pan, Duo, Col, Int, LAG, RAG, Rec, Bla, LFH, RFH
FLARE22	40	5	Liv, RKid, Spl, Pan, Aor, IVC, RAG, LAG, Gal, Eso, Sto, Duo, LKid
CHAOS	40	—	Liv
BTCV	30	—	Spl, RKid, LKid, Gal, Eso, Liv, Sto, Aor, IVC, PVSV, Pan, RAG, LAG
RAOS[3]	—	40	Liv, Spl, LKid, RKid, Sto, Gal, Eso, Pan, Duo, Col, Int, LAG, RAG, Rec, Bla, LFH, RFH, Pro, SV

Test Data Creation Different from existing work that solely chases for a higher segmentation accuracy, in this paper, we expect to evaluate the segment model's performance in dual tasks: The free-form text understanding ability and segmentation ability.

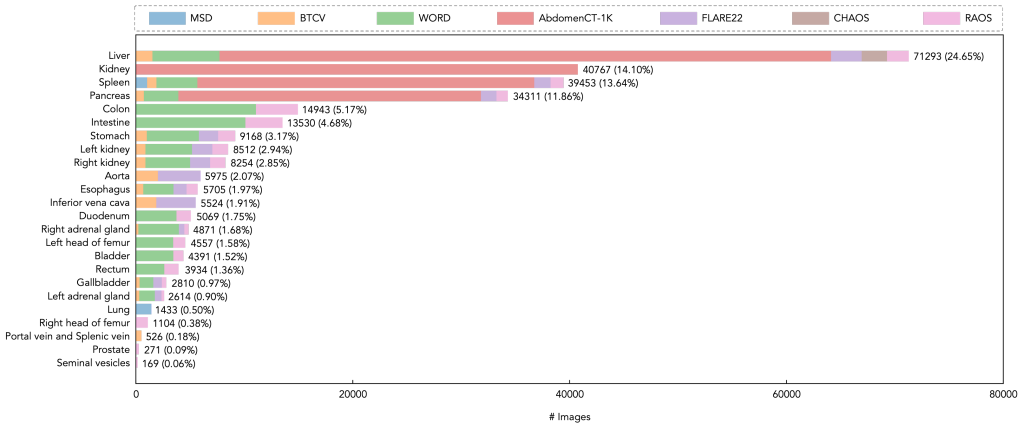

Figure 9: Distribution of labeled organs across the collected datasets. The image count for each organ and its corresponding ratio is marked in the plot.

In order to verify the model's ability to understand the language descriptions, we construct a query dataset (test set) from two resources: 1. Real-world human queries; 2. LLM-generated synthetic queries. For the first kind of real-world queries, we have two groups of annotators, **Domain Expert** and **Non-Expert**. Domain experts are from clinical hospitals who provide the query materials from their daily diagnosis notes, this group of people tends to use professional vocabulary, and their intention might not be explicitly expressed in a professional report, such as in the report, the doctor writes 'Concerns in the hepatic area that warrant a more focused examination', which implicitly means the 'liver is the area of interest under certain symptom'. Another group of query providers is the non-expert, who are not specialized in clinical or equipped with medical specialties. We explain to this group of people that their task is to write a sentence and show the intention of segmenting the target organ/tissue in a CT scan, e.g., the liver. This aspect of real queries represents a more general and non-specialist approach to expressing the need for segmentation (such as in the student learning scenarios). Apart from real query data, we incorporate synthetic test queries to enlarge the test samples and add randomness in various expressions. The synthetic test is generated by GPT-4o following the template shown below:

The Prompt Template to Generate Synthetic Queries.

System Description: You are a doctor with expert knowledge of organs.

Task Description: Now you are making a diagnosis of a patient on the CT scan over {body part}. You find a potential problem on {organ name} and want to see more details in this area, please query for segmentation by free-form text. Please make sure to deliver the segment target explicitly, and you are encouraged to propose various expressions.

Format: {segmentation query}, {explain reason}.

Example: Given that, {body part} is `abdomen` and {organ name} is `liver`.

[1]For simplicity, the following abbreviations are used: Liv (liver), Kid (kidney), Spl (spleen), Pan (pancreas), Col (colon), Int (intestine), Sto (stomach), LKid (left kidney), RKid (right kidney), Aor (aorta), Eso (esophagus), IVC (inferior vena cava), Duo (duodenum), RAG (right adrenal gland), LHF (left head of femur), Bla (bladder), Rec (rectum), Gal (gallbladder), LAG (left adrenal gland), RHF (right head of femur), PVSV (portal vein and splenic vein), Pro (prostate), and SV (seminal vesicles).

[2]Only the lung and spleen subsets from MSD were used.

[3]We used CancerImages (Set1) from RAOS as our out-of-domain test set. To avoid overlap, any scans in RAOS that were extended from WORD were excluded from testing.

Your response should be something like: {Please identify the liver for me for more analysis.} {Because elevated liver enzymes alanine aminotransferase (ALT) in the blood tests might indicate liver inflammation or damage}.

Output: {Placeholder}

The overall structure of the test dataset is shown in Figure 10. It consists of 25% expert queries, 25% normal queries, and half synthetic queries. In total, we have 2880 (24 organs x10 queries x3 x2x2) text queries. Each of the queries is labeled with the correct organ name to segment. This will be used to evaluate the ability of our learned TextEncoder model to understand correct intentions based on free-form language description.

At the same time, the organ names are connected to another segmentation test set, which contains several (how many) medical images such as CT scans, MRIs, etc. And stand on the results of interest-category identification, we conduct further segmentation result analysis, including the normal segmentation precision study, and also the equivariant identified segmentation study.

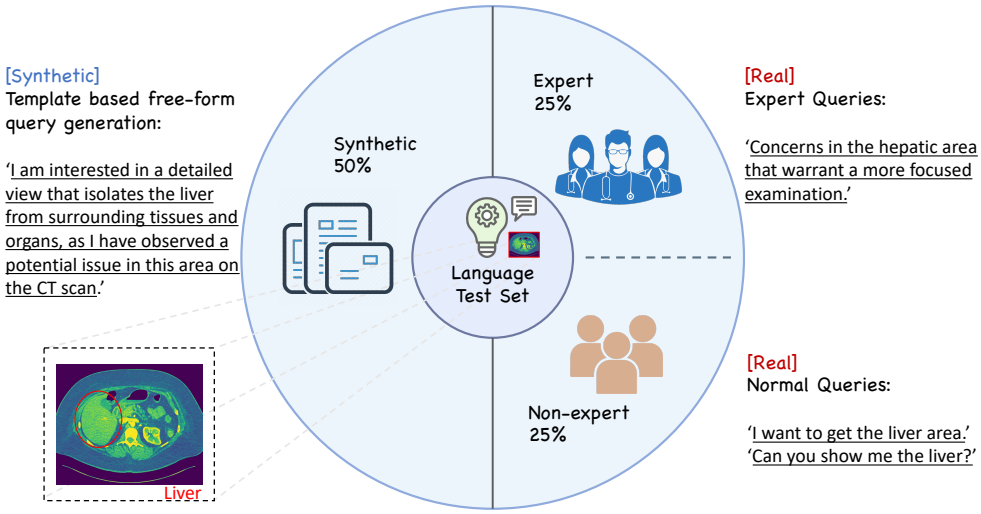

Figure 10: The Language Test Set for Verifying the Query Understanding Ability. It contains three aspects of components, real data - expert group, real data - non-expert group, and synthetic data.

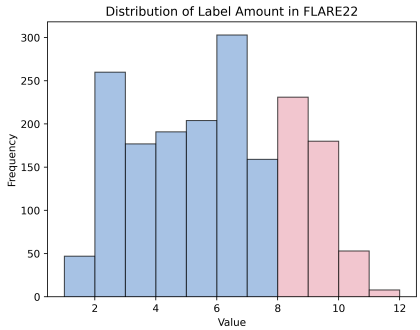

Figure 11: Positional prompt dataset provider split, we take the slices with more than α labels, where we set $\alpha = 8$ in this illustration (while 13 is the total label amount) as a split threshold, ensure that the slice used for training the label-agnostic provides sufficient semantics in the image content, such as left, upmost or largest, etc. Similarly, we process the other datasets such as BTCV and WORD.