Universal 3D Shape Matching via Coarse-to-Fine Language Guidance

Supplementary Material

6. Implementation Details

6.1. Details of MLLM prompting

After class-agnostic segmentation, we first render 3D objects with 3D masks into 2D images, by a front view and a back view, as shown in Fig. 8. We discard too small masks, *i.e.*, less than 5% pixels out of the whole object. This ratio is selected empirically to avoid misleading GPT-5. Similar to Find3D [40], we later feed the images into GPT-5 and the following prompt to obtain part names:

```
Infer region names
- What is the name of the part that is masked as [COLOR]? If you cannot find the part visible or are not sure, just say unknown. Only output one word or one phrase.
```

Finally, we aggregate part names into the 3D domain using known camera parameters to obtain the final semantic region proposal. As shown in Fig. 8, although the same concept can be expressed in different ways (*e.g.*, body and torso), our method can flexibly match them.

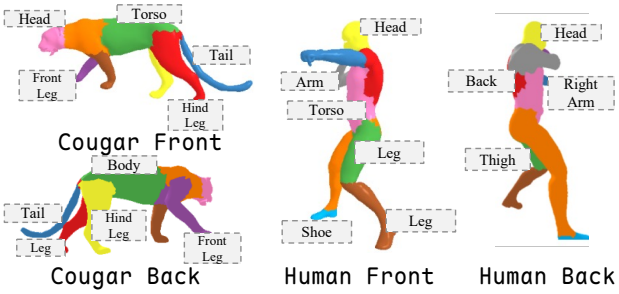

Figure 8. Examples of mask rendering and part names obtained from GPT-5.

6.2. Details of UniMatch

Class-agnostic segmentation. We consider an effective class-agnostic 3D segmentation method, PartField [34] to segment the 3D shape into non-intersecting parts. The number of semantic parts $n_{\mathcal{R}}$ is selected empirically to provide enough semantic information and avoid over-segmentation. We use $n_{\mathcal{R}} = 9$ for human data and $n_{\mathcal{R}} = 8$ for animal data. We also illustrate the segmentation results from PartField [34] and text-prompted method SATR [2] in Fig. 9. Compared with PartField, the segmentation results of SATR contain significant noise and ambiguous boundaries.

Semantic feature fields. We first perform view-consistent texture synthesis, SyncMVD [36], for uncolored shapes and render them into K multi-view RGB images using PyTorch3D [48] ($K = 10$ for all experiments). We use uniformly distributed elevation and azimuthal angles in $[0°, 360°)$. Then we extract semantic features using SD-DINO and employ the FeatUp upscaler [19] to upsample them. Lastly, we back-project 2D semantic features into the 3D domain from all visible views using known camera parameters and average them to obtain per-vertex features.

Figure 9. Comparison between class-agnostic segmentation PartField [34] and text-prompted segmentation SATR [2]. Different parts are rendered in random colors.

Network and functional map. We use DiffusionNet [53] as our feature refiner $\mathcal{F}$. The dimensions of $\boldsymbol{f}_{\text{geo}}$ and $\boldsymbol{f}_{\text{sem}}$ are 128 and 768 respectively. The output dimension of $\boldsymbol{f}_{\text{out}}$ is 256. For functional map calculation, we empirically set $\lambda_{\text{couple}} = 1.0$ and $\lambda_{\text{reg}} = 1.0$ in Eq. (1), following URSSM [9].

Training. We use the AdamW [38] optimizer with learning rate 10^{-3} for all experiments. We train our framework for 15 epochs.

7. Details of Datasets

7.1. Inter-Class Datasets

- **SNIS** [1] is a dataset to test strongly non-isometric and cross-category matching (*e.g.* a human *vs.* a dog), containing 211 shapes from different sources, including FAUST [7] (human), SMAL [65] (animals), and DeformingThings4D (DT4D) [29] (humanoid). Annotations include 34 semantically corresponding keypoints for each

shape. We use 250 test pairs provided by SNIS for evaluation (all test pairs are cross-category) and other pair combinations for training. Fig. 10 shows some example pairs from SNIS.

- **TOSCA** [8] consists of synthetic high-resolution meshes of humans and animals captured in different poses. It contains 80 objects that span several categories, including cats, dogs, horses, centaurs, gorillas, humans, *etc*. We remesh the original meshes to keep about $10,000$ faces. Since no cross-category annotations are provided, we prompted GPT-5 to find semantically matched keypoints between categories and finally filtered 20 keypoints. Similar with SNIS, we randomly select half samples from each category, and build the test pairs cross categories, result in 380 pairs for evaluation. We use the rest pair combinations for training.
- **SHREC07** [20] consists of 400 watertight shapes and 20 semantic categories with 20 shapes per class, representing a wide range of objects such as humanoids, mechanical parts, animals, and furniture. Since some categories contain extremely sparse keypoints and share almost no common structures, *e.g*. table, vase, and fish, we only conduct experiments on filtered categories, including human, teddy, armadillo, and four-legged animals, and prompted GPT-5 to find semantically matched keypoints between them, finally obtaining 15 keypoints. Similar to TOSCA, we randomly select half samples from each category, and build 100 pairs for evaluation and the rest pair combinations for training.

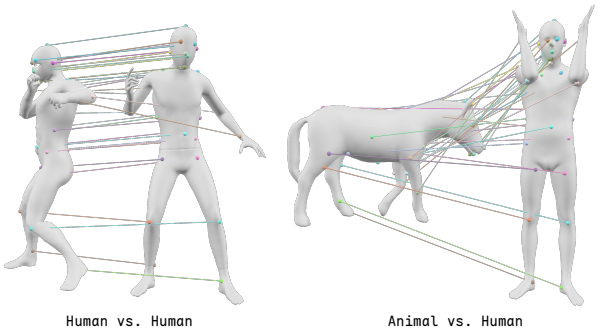

Figure 10. Example shape pairs and annotations from SNIS [1]. SNIS contains strongly non-isometric and cross-category shape pairs, such as human vs. animal and human vs. humanoid, with semantically annotated keypoint correspondences.

7.2. Non-Isometric Datasets

- **SMAL** [65] contains 49 four-legged animal shapes representing 8 species, such as dogs, cats, lions, horses, and *etc*. Each shape is represented as a watertight mesh with about 8,000 faces. Following Cao et al. [9], we use five species for training and three unseen species for evaluation.
- **TOPKIDS** [26] is designed for evaluating non-isometric deformations and topological noises. It contains 26 shapes of synthetic human ("fat kid") in various poses. Each shape includes ground-truth dense correspondences to a reference template shape.

7.3. Near-Isometric Datasets

- **FAUST** [7] consists of 10 human subjects with 10 different poses, where 80 shapes are provided for training and 20 shapes for testing. Following Cao et al. [9], we consider the remeshed versions from Donati et al. [13]. Each shape contains around 5,000 vertices with ground-truth dense correspondences to a template shape.
- **SCAPE** [4] contains 71 human shapes with various poses of the same person. The dataset is split into 51 training shapes and 20 testing shapes.
- **SHREC19** [42] is a benchmark to evaluate non-rigid shape matching, with a particular focus on different mesh connectivities in human shapes. It contains 44 human shapes with various identities and poses.

8. Details of Baselines

For URSSM [9], we use the same training scheme and parameter settings with UniMatch. *To keep a fair comparison, we disable the test-time adaptation.* For Diff3F [14], we follow the original implementation. Since the code of ZSC [1] is not open-source, we implement it according to the original paper. We use Grounding-DINO [35] for visual grounding and SAM [25] to obtain masks. Given segmentation results, the final dense correspondences are calculated by BCICP [49]. For DenseMatcher [63], we use the same training scheme, parameter settings, and the same semantic features on textured meshes with UniMatch. Since DenseMatcher requires predefined semantic parts, we use the same segmentation results of ZSC.

9. Limitation and Future Work

While UniMatch demonstrates strong performance in universal 3D shape matching under non-isometric and inter-class scenarios, several limitations remain. First, GPT-5 may fail to correctly order symmetric or repetitive parts (*e.g*., chair legs), as semantic part names alone cannot infer geometric arrangement. Incorporating explicit object orientation cues or relational priors may alleviate such issues. Second, our method still needs a separate procedure to extract semantic features using vision foundation models. Training a unified feed-forward feature extractor that distills visual knowledge from foundation models, rather than a feature refiner, may address this issue. Finally, the current framework relies on multimodal large language models for prompting during training-time part naming to en-

able semantic awareness. Extending this to more efficient pipelines could improve scalability and practicality. Future work will address these challenges by exploring enhanced structural priors of shapes, a unified feed-forward feature extractor, and reducing model intervention during matching.