HyperPredict: Estimating Hyperparameter Effects for Instance-Specific Regularization in Deformable Image Registration.

Aisha L. SHUAIBU   HTTPS://ORCID.ORG/HTTPS://ORCID.ORG/0000-0002-6061-2231
University of Sussex, Predictive Analytics Lab (PAL)

a.shuaibu@sussex.ac.uk

Ivor J. A. SIMPSON   HTTPS://ORCID.ORG/0000-1111-2222-3333
University of Sussex, Predictive Analytics Lab (PAL)

i.simpson@sussex.ac.uk

Abstract

Methods for medical image registration infer geometric transformations that align pairs, or groups, of images by maximising an image similarity metric. This problem is ill-posed as several solutions may have equivalent likelihoods, also optimising purely for image similarity can yield implausible deformable transformations. For these reasons regularization terms are essential to obtain meaningful registration results. However, this requires the introduction of at least one hyperparameter, often termed λ, that serves as a trade-off between loss terms. In some approaches and situations, the quality of the estimated transformation greatly depends on hyperparameter choice, and different choices may be required depending on the characteristics of the data. Analyzing the effect of these hyperparameters requires labelled data, which is not commonly available at test-time. In this paper, we propose a novel method for evaluating the influence of hyperparameters and subsequently selecting an optimal value for given pair of images. Our approach, which we call Hyper-Predict, implements a Multi-Layer Perceptron that learns the effect of selecting particular hyperparameters for registering an image pair by predicting the resulting segmentation overlap and measures of deformation smoothness. This approach enables us to select optimal hyperparameters at test time without requiring labelled data, removing the need for a one-size-fits-all cross-validation approach. Furthermore, the criteria used to define optimal hyperparameter is flexible post-training, allowing us to efficiently choose specific properties (e.g. overlap of specific anatomical regions of interest, smoothness/plausibility of the final displacement field). We evaluate our proposed method on the OASIS brain MR standard benchmark dataset using a recent deep learning approach (cLapIRN) and an algorithmic method (Niftyreg). Our results demonstrate good performance in predicting the effects of regularization hyperparameters and highlight the benefits of our image-pair specific approach to hyperparameter selection.

Keywords: Deformable Image Registration, Hyperparameter Selection, Regularization.

1. Introduction

Deformable image registration has been an active field of research for decades, as it is a fundamental process utilized in various medical imaging studies. The process involves aligning images of different modalities and time points for comparison and measurement of changes over time. It is very common for medical images to vary in their spatial resolution and orientation; as a result, non-linear image registration plays a vital role in various clinical

applications such as image-guided treatment delivery, pre-operative/post-operative assessment comparison, disease monitoring, disease diagnosis, and population analysis (Meng et al., 2022).

In medical images, the quality of registration largely depends on two conditions: (a) accurate alignment of anatomical structures, (b) a smooth and plausible displacement field. Achieving a balance between these factors is accomplished by utilizing hyperparameters that act as a trade-off between both objectives. Hence, selecting optimal hyperparameter values is a critical aspect when evaluating image registration methods.

The most quantitative method of selecting these hyperparameters is by performing grid or random search on the validation data using a discrete set of hyperparameter values. In this approach, additional data containing anatomical annotations is employed. By computing segmentation overlap of anatomical regions between the ground truth labels and the inferred registration, an "optimal" value is selected based on the hyperparameter that maximizes a criteria *across the entire population*. We argue that this approach may lead to sub-optimal choice as data-specific optimal hyperparameter differ significantly depending on various factors such as, the morphological similarity between the images and the anatomical structure of interest, among other factors. Furthermore, researchers may be inclined to adopt values from existing literature that may not be suitable for their specific dataset or registration task, equally leading to sub-optimal results.

In this paper, we present a novel method, HyperPredict, an efficient tool for instance-specific optimal hyperparameter selection in registration tasks. Our proposed approach enables greater flexibility at test time and can be leveraged in scenarios where labelled scans are scarce. In summary our contributions are as follows:

- We propose a novel and efficient method of *learning* the effect of registration hyper-parameters. This is achieved by learning the parameters of a network that maps a set of hyperparameters and image pair to desired evaluation metrics (derived from segmentation overlap and smoothness of deformation field).

- Our method is flexible and allows the efficient choice of optimal parameters based on a defined criteria.

- We test our proposed method using two different registration algorithms (cLapIRN and Niftyreg), our code is publicly available at `https://github.com/aisha-lawal/hyperpredict`.

The structure of the paper is as follows. Section 2 introduces deformable image registration, it's mathematical formulation and hyperparameters in medical image registration tasks, Section 3 describes related work. In Section 4 we present our method. Sections 5 and 6 describe experimental results and discuss insights. Finally, we present limitations of our method and conclude in Sections 7 and 8 respectively.

2. Background

Deformable Image Registration: Conventional registration algorithms (Ashburner, 2007; Beg et al., 2005; Avants et al., 2008b; Modat et al., 2014) solve the optimization

problem for each volume pair by iteratively improving estimates for the desired transformation such that the loss function is minimized. However, solving a pairwise optimization problem is computationally expensive and can be very slow in practical medical applications. Over the years, deep learning image registration methods (DLIR) (Mok and Chung, 2020b, 2021a; Balakrishnan et al., 2019) have been introduced to circumvent this by optimizing the parameters of a network, the general process aims to establish a dense, non-linear correspondence between pairs of images, such as magnetic resonance (MR) images or computerised tomography (CT) scans, by learning the optimal spatial transformation between the image pair that enhances similarity. The process of registering two images can be formulated as an optimization problem generally expressed as:

$$\phi^* = \arg\min_{\phi} \mathcal{L}_{\text{sim}}(f, m \circ \phi) \tag{1}$$

Equation 1 seeks to minimize a loss that consists of two parameters, f and $m \circ \phi$. Where m and f denote the moving and fixed image respectively, ϕ represents the deformation field, ϕ^* is the optimal registration field that maps the pixels/voxels from m to f.

Intuitively, deformable image registration poses a significant challenge due to its inherently ill-posed nature. Reliance solely on surrogate measures such as image similarity can be insufficient as they do not have the ability to differentiate between accurate and inaccurate registrations (Rohlfing, 2011). For example, given an image pair, m and f, registration algorithms seek to find the transformation that deforms the moving image, mapping the coordinates of m to f; however, ground-truth deformation fields do not exist to serve as a reference point, and without any restrictions placed on the transformation properties, the cost function becomes poorly conditioned. Thus, to ensure tractability, registration algorithms employ some regularization that imposes a constraint on the estimated deformation field - reducing the set of possible solutions. The quality of the registration largely depends on the regularization weight, λ, that serves as a trade-off between the quality of the registration and how smooth the deformation field is. Thus, the cost function can be reformulated as follows:

$$\phi^* = \arg\min_{\phi} \mathcal{L}_{\text{sim}}(f, m \circ \phi) + \lambda\mathcal{L}_{\text{reg}}(\phi) \tag{2}$$

Description of the notation is similar to Equation 1, with λ denoting the regularization weight, $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{reg}}$ represent the dissimilarity and imposed regularization function respectively. The aim is to optimize both the registration quality, $\mathcal{L}_{\text{sim}}$, while having a smooth deformation field, $\mathcal{L}_{\text{reg}}$. The choice of loss function for $\mathcal{L}_{\text{sim}}$ depends on various factors including the intensity distribution and contrast of the image. Commonly used functions are mean squared error, mutual information (Viola and Wells III, 1997), and normalized cross correlation (Avants et al., 2008a), which capture different aspects of similarity. For $\mathcal{L}_{\text{reg}}$, a diverse range of regularisers can be employed to enforce spatially smooth deformations, examples include, linear elasticity, bending energy, and learned priors.

Evaluation Metrics: Quantifying the accuracy of non-rigid image registration is inherently difficult. As such, no gold standard for evaluating deformable registration exists. However, with *additional labelled data*, we can make an effort to quantify the quality of the

registration with some degree of confidence. A common approach as a means of evaluation is to compute the segmentation overlap of different anatomical regions between m and f. The Dice score (Dice, 1945) is one way to do this. While the Dice score between anatomical structures is a reliable proxy for evaluation, a high Dice score does not imply a biologically plausible registration. This is because a deformation field with overlapping voxels can still yield a high Dice score despite potentially unrealistic deformations (Rohlfing, 2011). Therefore, in addition to the Dice score, assessing the deformation's diffeomorphic property is equally important in order to preserve the topology of the features in the transformed image. Hence, the determinant of the Jacobian $|J_\phi|$ serves as a measure of smoothness of the deformation field.

The experiments presented in this paper employ the Dice score and number of folded voxels (derived from $|J_\phi|$) as the desired evaluation metrics. However, it is worth emphasizing that our methodology is not limited to these specific choices. Our approach is flexible and can be easily extended to accommodate a wide range of hyperparameters and evaluation metrics. Depending on specific application or research objective, our method can be adopted to fit the desired metrics, such as Hausdorff distance and Target Registration Error (TRE) amongst others.

Hyperparameter Selection: Hyperparameter optimization algorithms tackle the challenge of jointly optimizing both model hyperparameters and model weights through a validation and training objective respectively. The simplest approach involves treating model training as a "block-box" function and employing methods such as grid search, random search, or sequential search (Bergstra and Bengio, 2012). Other methods include manual fine-tunning and Bayesian optimization (Bergstra et al., 2011; Turner et al., 2021; Snoek et al., 2012; Mockus, 1998). Although effective, these methods can be inefficient as they require repetitive optimization procedures for each hyperparameter value. We describe existing methods in Section 3.

Hyperparameters in Registration: In non-rigid image registration, the number of hyperparameters to be optimized depends on the specific registration algorithm and objective. For example, cLapIRN, (Mok and Chung, 2021a) regulate the smoothness of the deformation field using a *single* registration hyperparameter, denoted as λ. On the other hand, algorithms such as ConvexAdam, (Siebert et al., 2021) and Niftyreg, (Modat et al., 2014) employ a *set of* hyperparameters to optimize the registration process. In Niftyreg, spacing for spline interpolation, bending energy, and linear elasticity are some of the regularization options to govern the diffeomorphic properties of the deformation field. In such scenarios, especially when dealing with algorithms with high computational complexity, employing the above methods to tune multiple parameters becomes impractical.

Motivated by the challenge above, HyperPredict presents a more efficient method for selecting optimal hyperparameter values. During training, HyperPredict learns the effect of the hyperparameter on the evaluation metrics. By utilizing a registration algorithm, the target values of both metrics (described above) are obtained for backpropagation. At test time, given an input $\{m, f, \lambda\}$, and without having true segmentation, the model predicts the metrics associated with the input. Based on a specific criteria (defined in method section), we select optimal parameter, λ^*, and use that for registration.

3. Related Work

As described in Section 2, registration methods typically optimize a data term and a weighted regularization term. Hence, to be able to minimize the error of the registered results, it becomes crucial to optimize the independent hyperparameters of the registration model. Despite the advantages that come with DLIR methods, they still face challenges in navigating the trade-off parameter within the objective function. Traditionally, the selection of regularization parameter values relied on a trial and error approach (Ashburner, 2007; Andersson et al., 2007; Rueckert et al., 1999). In this, users manually search for parameters that yield satisfactory results for a given dataset. This process often involves fine-tuning multiple parameters, which can be time consuming, particularly in hierarchical registration methods. An example is studies conducted in (Rühaak et al., 2017), their registration method utilized four distinct parameters. To capture the effect of each parameter, they conducted separate experiments, in each, one parameter was varied while the other three remained fixed. This iterative process allowed them to empirically determine an "optimal" value for all four parameters. Although ad hoc parameter-tuning may produce satisfactory outcomes, it requires expert domain knowledge to effectively guide the tuning process (Joshi et al., 2004; Vialard et al., 2011; Ma et al., 2008; Wang et al., 2019).

Cross-validation is another standard practice employed for selecting registration hyperparameters (Balakrishnan et al., 2019; Mok and Chung, 2020a). Using grid search, the parameters that yield the best performance on a given dataset is selected for subsequent registrations. As a result, it utilizes a *fixed* level of regularization across the entire set of image pairs, with the assumption that all image pairs require the same degree of regularization. Additionally, this process necessitates availability of manually labelled dataset that accurately represents the testing data of interest. This method of parameter selection requires substantial computational resources and human effort, which may result in sub-optimal parameter choices.

Employing a hierarchical Bayesian model to infer the regularization parameter is another method demonstrated in studies by (Risholm et al., 2012) and (Simpson et al., 2012). The general strategy of Bayesian approach leverages a probabilistic model to explore and evaluate the performance of hyperparameters that result in improved registration. This is achieved by characterising the posterior distribution using techniques like Markov Chain Monte Carlo (MCMC) or Variational Inference. (Risholm et al., 2013) considered scenarios where the confidence of both observed data and model priors are unknown, aiming to tackle the challenge of finding an objective trade-off between the two terms. They characterised the posterior distribution using Metropolis-Hastings and treated the hyperparameters as latent variables approximately marginalized over. A similar approach is proposed in (Wang and Zhang, 2021; Zhang et al., 2013). Given sufficient samples, MCMC yields a good characterization of the posterior, however, the computational demands and complexity of Markov chain is a restricting factor that limits the feasibility of this approach. While Variational Bayes(VB) has less computational burden, it may compromise on the quality of estimates. (Simpson et al., 2015) utilized a spatially adaptive prior to limit unwanted regularization on the estimated transformation. One limitation is that the uncertainty estimates quantify the variability in the displacement field based on the inferred hyperparameters, but they do not

690

account for the uncertainty in the registration caused by variations in the hyperparameters themselves (Le Folgoc et al., 2016).

Recent advancements in deep learning registration methods such as HyperMorph, (Hoopes et al., 2021) and cLapIRN, (Mok and Chung, 2021b) propose an approach that eliminates the need for repeatedly training models in search of hyperparameters that boost model performance. They do this by learning the effect of hyperparameters on the deformation field. Specifically, cLapIRN proposes a conditional image registration method that learns the conditional features that are strongly correlated with specific regularization values. On the other hand, HyperMorph leverages a secondary network to generate conditioned weights for the primary network, in order to learn the impact of registration hyperparameters on the deformation field. To quantify the choice of hyperparameter (selected arbitrarily), registration methods require labels. Rather than learning the entire registration process, (Niethammer et al., 2019), focus on learning a spatially adaptive regularizer within a registration model to preserve the desired level of regularity, however, this is not done on a pair-wise basis. A similar method is adapted in (Vialard and Risser, 2014) using a learning based approach.

To summarize the challenges above; depending on the selected method, tuning and selection of optimal hyperparameters involves dealing with one or more of the following (a) computational expense, (b) time consuming nature of the process, (c) requires labelled data at test time which involves human effort and may not readily available, (d) ineffective in selecting optimal hyperparameter (assumes a fixed value throughout the entire dataset). While both amortized inference and algorithmic approaches to registration allow for hyperparameter selection at test time, efficiently selecting an optimal value at test time remains a challenge, which we aim to solve with our proposed HyperPredict.

We emphasize that the goal of HyperPredict is not to function as a registration tool or to directly compare it with existing ones, but rather to serve as a means of aiding registration algorithms in selecting appropriate hyperparameters.

4. Method

Current unsupervised learning-based image registration methods that learn the effect of hyperparameters define a network $h_\psi(m, f, \lambda) = \phi$, where h_ψ is parameterized by a convolutional neural network (CNN). Our presented approach takes advantage of these registration algorithms but uniquely learns a function g_θ that captures the correlation between a hyperparameter value and the evaluation metric for an image pair. Given e_o(an encoded representation of m and f) and sampled hyperparameter value, λ (drawn from a log-normal distribution, i.e., $\lambda \sim \text{LogNormal}(\mu, \sigma)$), we parameterize our proposed method as a function, $g_\theta(e_o, \lambda)$ with a Multi-Layer Perceptron (MLP). The proposed method works with any registration algorithm that incorporates a regulariser in its smoothness during optimization.

Figure 1 presents an overview of our method. The network architecture during training is comprised of an encoder, a registration algorithm, and the MLP. At train time, we freeze the pretrained weights of the encoder and registration algorithm. Based on the encoded representation and hyperparameter passed as input, the MLP learns to predict the Dice coefficient and number of folded voxels as the evaluation metrics. We use the deformation field, ϕ, from the registration algorithm and the Dice score between the warped moving label, $s(m) \circ \phi$, and fixed label, $s(f)$, to enable the computation of the loss. At test time,

given unseen images and regularization weight, $\{m, f, \lambda\}$, we obtain the predicted metrics by evaluating $g_\theta(e_o, \lambda)$. All notations are illustrated in Figure 1.
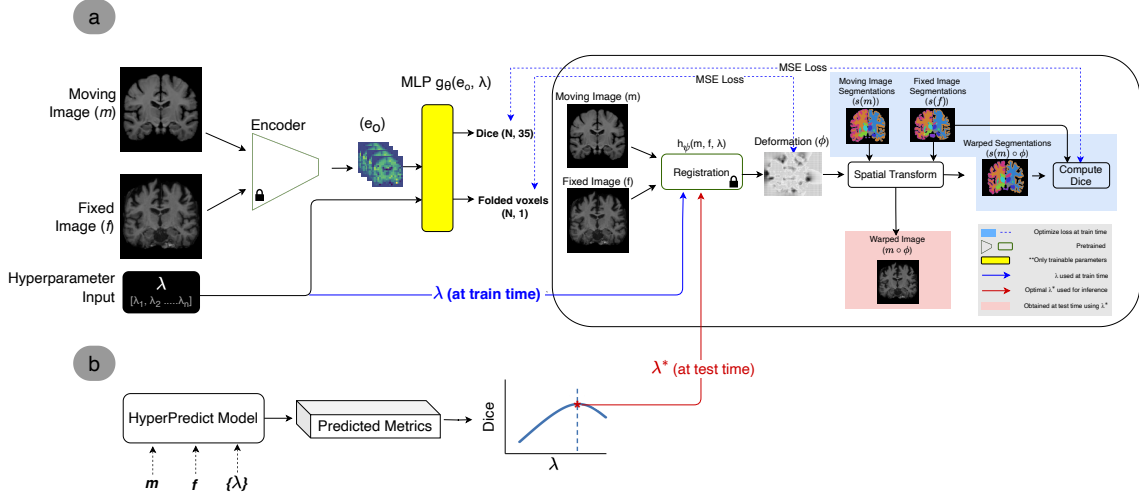

Figure 1: Overview of the method: **(a)** We learn the parameters g_θ (**yellow box**) of an MLP that maps the encoded representation (e_o) of an image pair (m, f) and a set of hyperparameters (λ) to evaluation metrics (Dice and folded voxels). At train time, we optimize g_θ by computing the target metrics of the input (**blue boxes and dashed blue lines**) using a registration algorithm. **(b)** At test time, given a set of input, we predict the effect for each hyperparameter value. We select the optimal hyperparameter, λ^*, and use that in the registration algorithm to obtain a desired warped image (**red box**). The green shapes (encoder and registration) are pretrained.

When inferring the optimal weight, λ^* (as in Figure 1(b)), we can impose selection criteria. For instance, we can consider only results that restrict significant discontinuities in the deformation field. This is achieved by constraining the percentage of number of folded voxels (nfv) allowed. Based on this constraint, the optimal weight, λ^*, that corresponds to the highest Dice score is obtained (across image pairs or selected labels). Mathematically, this can be expressed as;

$$\lambda^* = \arg\max_\lambda (\text{Dice}(\lambda)\mathbb{1}[\text{nfv} < \text{N}0.5\%]) \tag{3}$$

Equation 3 determines the optimal value, (λ^*) by maximizing the Dice score while ensuring %nfv is less than 0.5% of total folded voxels (N). The derived λ^* is used for registration. In the following sections, we describe our method in detail.

4.1 HyperPredict Architecture

This section describes the architecture used in our experiments. Let moving image, m and fixed image, f represent an image pair, each defined over a 3D spatial domain, $\Omega \subset \mathbb{R}^3$. Since

CNNs capture abstract representation of images (Szeliski, 2022), we leverage this to define our encoder. The encoder takes m and f as input and outputs an encoded representation, e_o, of the image pair. For simplicity, our experiments utilize two pre-trained registration models as encoders; Symnet, (Mok and Chung, 2020a) and cLapIRN, (Mok and Chung, 2021a). It is important to note that we need an encoder that could take in a pair of 3D-volumes, the conventional approach of dealing with this is by utilizing a registration model. We performed preliminary experiments using 2D implementation of masked autoencoder (He et al., 2022) to learn a self-supervised encoding, we found this very slow since they are not designed to work with 3D data.

The encoder is represented as $\mathbb{R}^{N,C_m,D_m,H_m,W_m} \times \mathbb{R}^{N,C_f,D_f,H_f,W_f} \rightarrow \mathbb{R}^{N,C_{e_o},D_{e_o},H_{e_o},W_{e_o}}$. With N, C, D, H, W, indicating the batch size, channels, depth, height and width respectively. The subscripts m, f, e_o, denote moving, fixed and encoded representation respectively. Using Symnet as an encoder, the resulting representation, e_o has dimensions of 1x56x10x12x14. We flatten the last three dimensions (D,H,W) and compute the mean across the channels. The result is concatenated with λ and passed as input to the MLP. We explored other summary statistics, such as concatenating the mean, max, and min across the channels, comparable results are presented in Appendix C.

Throughout our experiments, we parameterize the function $h_\psi(.,.)$ with two different registration algorithms; cLapIRN, and Niftyreg. Both of which accept hyperparameters as input that condition the smoothness of the registered image. At each iteration, the same set of image pair and hyperparameter is passed to the encoder and registration algorithm simultaneously, the loss is computed between the predicted values from the MLP and target values from the registration.

4.2 Loss Functions

In this section, we describe the loss function and evaluation metrics of the model. We optimize the MLP parameters, g_θ using stochastic gradient descent to minimize the loss. Mathematically, the objective function at each iteration is defined as;

$$L = \mathcal{L}_{\text{overlap}} + \alpha \mathcal{L}_{\text{nfv}} \tag{4}$$

$\mathcal{L}_{\text{overlap}}$ and $\mathcal{L}_{\text{nfv}}$ are the two components of the loss L, representing the Dice loss and nfv loss respectively. This is a multitask learning problem, hence we define α as a weighting term, balancing the relative magnitude of both losses, Appendix F shows the relative effect of training with different values of α. From the figure, 1.0 proves to be optimal for learning the metrics of both cLapIRN and Niftyreg registration methods. Both components of the loss are evaluated using the mean squared error defined in Equations 6 and 7 below.

Assessing the quality of registration is non-trivial due to competing objectives. Low regularization parameter can allow for close matching of appearance at the cost of anatomically-implausible and highly irregular deformations. Similarly, high value of smoothness regularization enables smooth deformations with sub-optimal alignment. In practice, amongst other evaluation metrics, the Dice score and $|J_\phi|$ are used to access the result of the registration. In our method, we focus on these two metrics.

Dice score: Acquiring accurate human annotated anatomical segmentations is a tedious task, hence the annotations available are leveraged during training. For a well aligned

image, regions in f and $m \circ \phi$ corresponding to the same anatomical region should overlap well. The Dice score quantifies the overlap between the structures. Let $s(f)_i$ and $s(m)_i$ represent labels of volumes f and m of structure i respectively, we define the target Dice score as;

$$\text{Dice}(s(f)_i, s(m)_i \circ \phi) = 2 * \frac{|s(f)_i \cap (s(m)_i \circ \phi)|}{|s(f)_i| + |s(m)_i \circ \phi|} = \hat{y}_i \tag{5}$$

Where a score of 1 indicates a perfect match between anatomical labels, and 0 means there is no overlap. Thus, we can mathematically define $\mathcal{L}_{\text{overlap}}$ as MSE between the predicted Dice, y_i and the target Dice, $\hat{y}_i$ given as;

$$\mathcal{L}_{\text{overlap}} = \frac{1}{|N|} \sum_{i \in N} (y_i - \hat{y}_i)^2 \tag{6}$$

Since we also aim to evaluate the effect of hyperparameter on each label, we predict 35 different Dice scores, each representing the degree of overlap for an anatomical region in the brain.

Number of folded voxels: We obtain the target number of folded voxels from the registration by computing the Jacobian determinant of the deformation field $|J_\phi|$ for each input, this gives us an understanding of its diffeomorphic properties. Where negative values of $|J_\phi|$ indicates non-plausible mapping, values less than 1 means contraction around that local region and values greater than 1 denotes expansion within that region. In practice, the displacement field is represented in an n + 1D image, meaning for each voxel p within the spatial domain, Ω, the displacement u(p) represents a shift that aligns f(p) and $[m \circ \phi]$(p) to corresponding anatomical regions. The target number of folded voxels, x_i, is derived by computing the sum where $|J_\phi| < 0$ acorss the entire displacement field. Hence, $\mathcal{L}_{\text{nfv}}$ in Equation 4 can be formally defined as;

$$\mathcal{L}_{\text{nfv}} = \frac{1}{|N|} \sum_{i \in N} (x_i - \hat{x}_i)^2 \tag{7}$$

Where x_i and $\hat{x}_i$ denote the predicted and target nfv respectively. Throughout our experiments, we express nfv as a percentage, i.e., %nfv.

5. Results

We evaluate our method on 3D brain MR scans. To demonstrate our contributions, we train and evaluate HyperPredict models using two pre-trained convolutional encoders and two registration models as previously mentioned. Subsequent sections provide a comprehensive description of the experimental setup and the experiments conducted.

Dataset and Pre-processing: We demonstrate our method on brain MRI registration task. We use 414 T1-weighted brain scans from the OASIS dataset (Brains, 2020; Marcus et al., 2007). The dataset contains cross-sectional collection of subjects (men and women) aged from 18 to 96. The pre-processed version of OASIS dataset is obtained from (Hoopes et al., 2021). They performed standard pre-processing steps using FreeSurfer, (Fischl, 2012) including; affine spatial normalization, skull stripping, sub-cortical structures segmentation,

and finally, cropping the resulting images to 160 x 192 x 224. The dataset includes sub-cortical segmentation maps of 35 different anatomical structures for each volume, generated by FreeSurfer. We consider Freesurfer a silver-standard method for generating automatic brain segmentations (Puonti et al., 2016). We used the resulting segmentation maps during training to compute the target segmentation overlap between the image pair. We divide this dataset into train, validation and test sets of sizes 254, 80, and 80 respectively.

Implementation: To improve the computational efficiency of our experimental process, we precompute registrations using both cLapIRN and Niftyreg between all images pairs using randomly selected regularization weights, sampled from a log-Normal distribution. We save the resulting metrics of these registrations in a csv file, which is used to train the MLP network.

The parameterization of g_θ is based on three linear layers, each followed by a LeakyReLU (Maas et al., 2013) activation function except the final layer. Our proposed method is implemented in PyTorch, we adopt Adam optimizer, (Kingma and Ba, 2014) with a learning rate of 10^{-4}. The training process follows the description in Section 4.1.

We ran our model using both the SymNet and cLapIRN encoders. The results presented in Appendix C demonstrates that SymNet achieved a lower Mean Absolute Error for both the Dice score and %nfv. Hence, for all experiments detailed in this section, we show results for Symnet as an encoder. To experiment HyperPredict's ability to generalize across different registration methods, we show results for HyperPredict on cLapIRN and Niftyreg registration methods separately. We label HyperPredict trained with both methods as HyperPredict$_{\text{clap}}$ and HyperPredict$_{\text{nr}}$ respectively. We denote cLapIRN *single* registration hyperparameter as λ. Niftyreg offers various regularization options, we use spacing, *sx*, bending energy, *be*, and linear elasticity, *le* in our experiments.

5.1 Experiment 1: Accuracy of HyperPredict

This experiment aims to validate our proposed method by assessing if, and how accurately a HyperPredict model is able to predict the desired evaluation metrics. We check to see if through the registration algorithms, HyperPredict learns to capture the effect of a wide range of hyperparameter values simultaneously. We present a validation plot that shows our ability to predict over both metrics for selected values of hyperparameters on the test dataset. Additionally, we present Bland-Altman plots to evaluate the agreement between predicted values from HyperPredict and the resulting registration metrics.

Experiment Setup and Results: First, we separately train two HyperPredict models, one for each registration algorithm (cLapIRN and Niftyreg) using the steps outlined in method section. To measure the models predictive performance, we analyze the residual plots between the predicted and target metrics. To do this, we sample arbitrary discrete values of hyperparameters; $\lambda \in [0.05, 0.075, 0.1, 0.125, 0.150, 0.2, 0.5, 1]$, and $be \in [0.0001, 0.001, 0.0075, 0.05, 0.075, 0.1, 0.125]$, we observed more variance with smaller values of bending energy. We use the sampled hyperparameter values to run image pairs in the test set on HyperPredict$_{\text{clap}}$ and HyperPredict$_{\text{nr}}$ respectively. Similarly, for each test pair and hyperparameter value, we derive the corresponding target metrics from both registration methods. Finally, for the range of hyperparameter values selected, we compute the difference
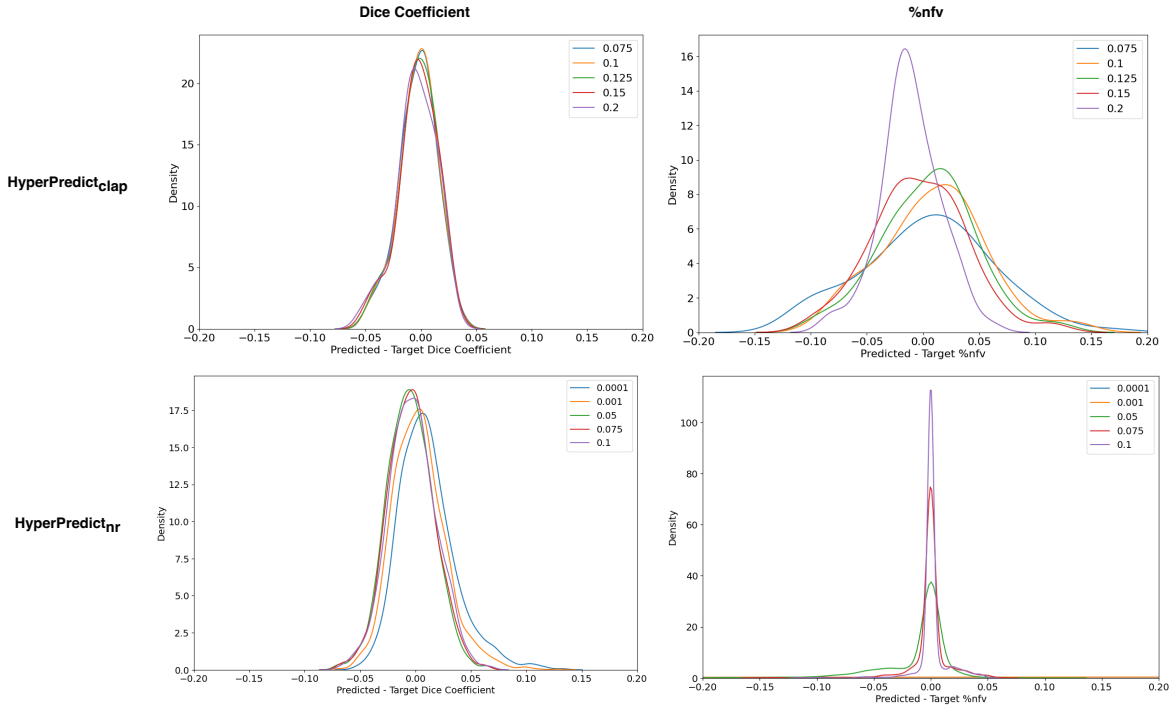
Figure 2: Error Distribution Plots on Both HyperPredict Models. **Left:** Difference in predicted and target dice coefficients for selected hyperparameter values. **Right:** Difference in predicted and target %nfv for selected hyperparameter values. For visualization purpose, we display results for selected values.

between the predicted and target for both metrics on each HyperPredict model. We present the error distribution plots in Figure 2.

To assess the agreement between the predicted and target metrics, we use Bland-Altman plots. For HyperPredict$_{\text{clap}}$, we generate 200 log-linearly spaced values between -10 and 0, $\lambda \in [-10, 0]$. Similarly, for HyperPredict$_{\text{nr}}$, we sample 200 log-linearly spaced values for the bending energy, $be \in [-10, 0]$, we use a fixed value of 5 for the spacing, $sx \in$ [5] (Segmentation overlap metrics indicate that the smallest spacing is better in Niftyreg, see Figure 9 (left) in Appendix A, but we compromise for computational efficiency), and a fixed value of 0.01 for linear elasticity, *le*. For image pairs in the test dataset, with HyperPredict$_{\text{clap}}$, we run each of the 200 sampled hyperparameter value, λ with the pair to obtain the predicted metrics. Next, we select the optimal λ value per image pair and pass it to the cLapIRN registration to obtain the target metrics. The same procedure is done for HyperPredict$_{\text{nr}}$, using sampled *be*, and fixed values of *le* and *sx* as parameters.

In Figure 3 we present the pairwise differences between the predicted and target metrics plotted against their average, for both HyperPredict$_{\text{clap}}$ (top) and HyperPredict$_{\text{nr}}$ (bottom). The plots reveal a close agreement between results from the predicted and target metrics for HyperPredict$_{\text{clap}}$, with narrow upper and lower boundaries, indicating that majority of the differences fall within an acceptable range. Furthermore, the figure demonstrates that
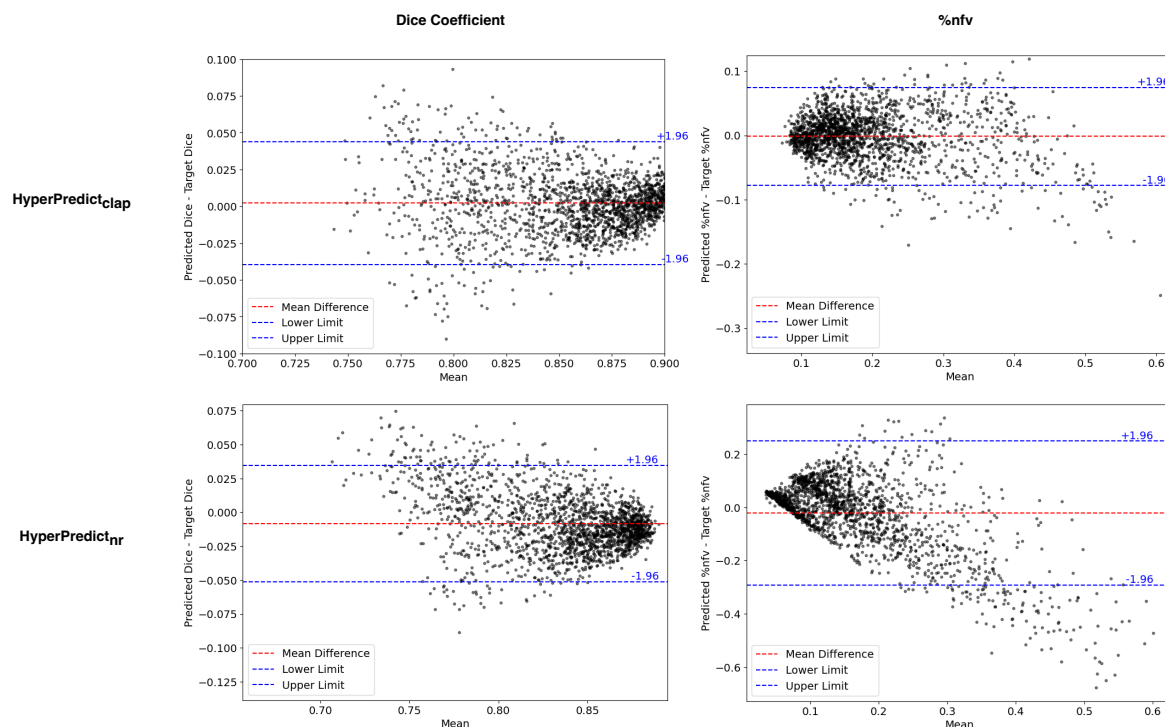
Figure 3: Validating HyperPredict. **Left:** Bland-Altman plots showing the agreement between the predicted vs target Dice scores across the entire population for both models (each point represents the average Dice for a particular image pair). **Right:** Bland-Altman plots showing the agreement between the predicted vs target %nfv for the entire population.

our method is able to yield metrics similar to those obtained through cLapIRN registration for each image pair. We found an average difference in the predicted and target Dice scores to be 0.015±0.014 and 0.028±0.026 for the %nfv.

Using HyperPredict$_{nr}$, we observe reasonable agreement between the predicted and target dice scores (bottom left), however, our model exhibits some limitations in accurately predicting %nfv (bottom right). The determined difference is 0.018±0.013 and 0.099±0.109 for Dice and %nfv respectively.

5.2 Experiment 2: HyperPredict vs Cross-Validation

The standard approach to obtaining an optimal hyperparameter is by performing cross validation. Hence, this section aims to evaluate the effect of inferring instance specific hyperparameters using HyperPredict. We achieve this by comparing the results obtained from our method with standard cross-validation.

Experiment Setup and Results: In this section, we conduct three experiments.

Step 1: In the first experiment, we compare the Dice scores obtained using our method on selected anatomical structures and %nfv with that of cross-validation. For HyperPredict$_{\text{clap}}$ and cross-validation on cLapIRN, we leverage the low computational burden offered by HyperPredict to test multiple hyperparameter values on a single pair. The sampling method is similar to the previous experiment. At test time, we run the sampled values for each subject in the test set through the trained HyperPredict$_{\text{clap}}$ network. Using the criteria defined in Equation 3, we select the optimal hyperparameters across specific labels. Finally, we register the image pair using the derived optimal values to obtain the corresponding target Dice coefficient and %nfv.
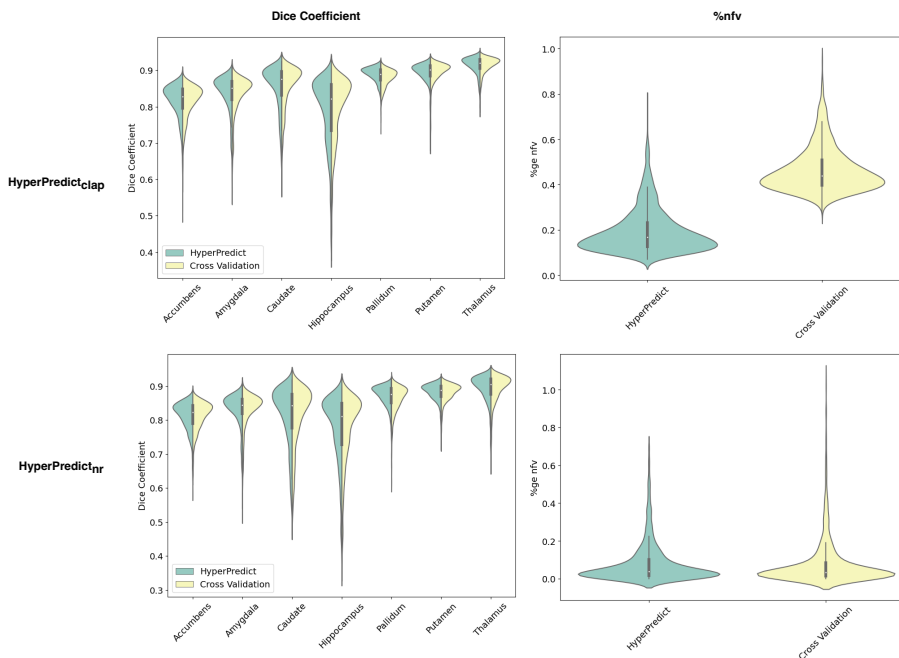
Figure 4: HyperPredict vs Cross-Validation. **Left:** Comparison of the Dice Coefficient on both methods for selected labels across the population. **Right:** Comparison of the %nfv on both methods across the population.

Since in practice it is not feasible to run 200 hyperparameter values per image pair on a registration algorithm in search of the optimal hyperparameter, we perform cross-validation using 8 discrete values of λ selected arbitrarily, $\lambda \in [0.05, 0.075, 0.1, 0.125, 0.150, 0.2, 0.5, 1]$, cross-validation on cLapIRN found optimal value, λ^*, to be 0.075. We use this for registration at test time across the *entire* test dataset.

We repeat steps the above steps using HyperPredict$_{\text{nr}}$ and Niftyreg registration algorithm. For cross-validation, we maintain a fixed spacing value of 5, a fixed linear elasticity of 0.01, and discretely sample the bending energy, $be \in [0.0001, 0.001, 0.0075, 0.05, 0.1, 0.2, 0.5, 1]$. Cross-validation on Niftyreg found optimal value, be, to be 0.0075.

We compare the results of HyperPredict and cross-validation for both experiments using the Dice coefficient scores and %nfv. In Figure 4 (top row), we present results of

HyperPredict$_{\text{clap}}$ vs cross-validation on cLapIRN for both metrics. The bottom row of the figure depicts results obtained from Niftyreg.

From the Figure, HyperPredict$_{\text{clap}}$ demonstrates comparable Dice scores with those obtained through cross-validation, with HyperPredict slightly outperforming cross-validation on a few anatomical structures. Interestingly, we find that our method of selecting optimal hyperparameter values leads to a reduction in folded voxels, indicating more desirable (plausible) registrations (top right). Similarly, the experiment with Niftyreg shows comparable dice scores between both methods, and the image pair with the highest %nfv has a value less than that of cross-validation.
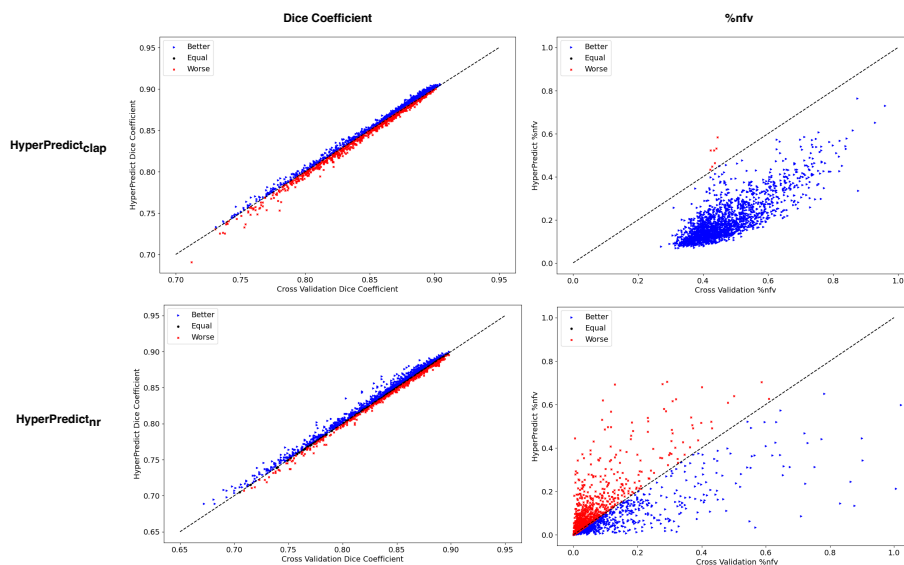
Figure 5: HyperPredict vs Cross-Validation. **Top:** Worse case analysis on HyperPredict$_{\text{clap}}$ **Bottom:** Worse case analysis on HyperPredict$_{\text{nr}}$.

Step 2: For the second experiment, we perform a best and worst case analysis in Figure 5, where we plot the value of the metric derived from HyperPredict against cross-validation for each image pair. We present results for HyperPredict$_{\text{clap}}$ and HyperPredict$_{\text{nr}}$. For all four plots depicted in the figure, the best cases are represented with a blue marker, indicating samples where HyperPredict out-performs cross-validation, red markers represent regions where cross-validation surpasses our proposed method, and equal cases are represented as black points that lie on the dotted slope.

Step 3: In the final experiment comparing cross-validation with HyperPredict, we run the registration of a single image pair using the optimal $\lambda^* = 0.075$ derived from cross-validation on cLapIRN. Similarly, we perform the registration on the same pair using the optimal $\lambda^* = 0.102$ obtained from HyperPredict$_{\text{clap}}$ for that specific pair. The outcomes of these two approaches are illustrated in Figure 6. We record the dice score prior to registration as 0.425. The dice scores of both methods after registration are 0.755 and 0.751 respectively, and their corresponding folded voxels as 0.73% and 0.48%.
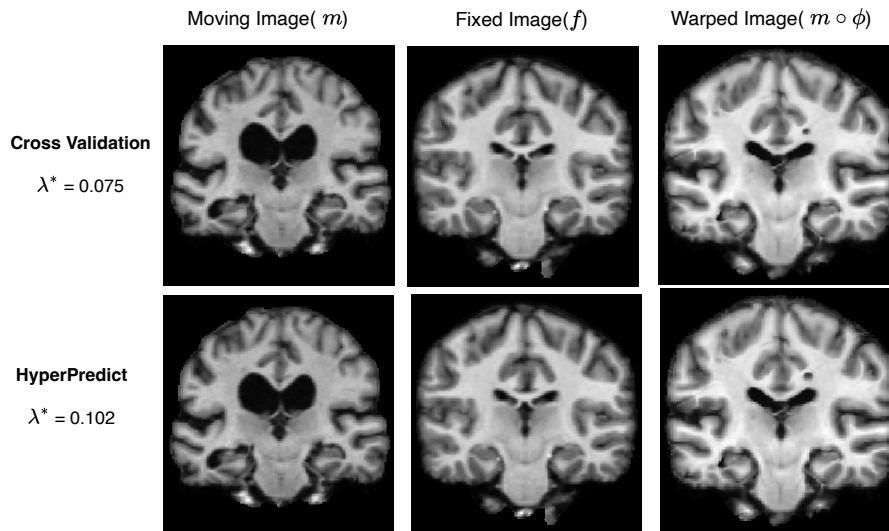
699

Figure 6: Cross Validation vs HyperPredict on cLapIRN registration: **Top:** Using optimal value of $\lambda^* = 0.075$ derived from cross validation to register an image pair. **Bottom:** Using optimal value of $\lambda^* = 0.102$ derived from HyperPredict$_{\text{clap}}$ to register the same image pair.

5.3 Experiment 3: Analyzing the Distribution of Hyperparameters

The objective of this experiment is to assess the distribution of regularization values for the two variants of HyperPredict: HyperPredict$_{\text{clap}}$ and HyperPredict$_{\text{nr}}$, and also identifying the best values to utilize for registration. We present the results obtained when optimizing for two instance-specific cases; across structures and across subjects. For visualization purposes, we present results of selected sub-cortical anatomical regions (Thalamus, Caudate, Putamen, Pallidum, Hippocampus, Amygdala, Accumbens).

Experiment Setup and Results: Maintaining the defined criteria and experimental setup in previous experiments, this section is subdivided into two distinct experiment.

Step 1: The first experiment seeks to validate our hypothesis that optimal values differ on a subject and structure basis. We infer the results of HyperPredict$_{\text{clap}}$ and HyperPredict$_{\text{nr}}$ using the test dataset and sampled values of λ and be respectively. From the result, we obtain the distribution of optimal hyperparameters across subjects (right) and selected labels (left) presented in Figure 7.

Step 2: In the second experiment we identify label specific optimal regularization weights. For an image pair, HyperPredict derives 35 different Dice scores, each for an anatomical region. We leverage on this to observe the effect of utilizing instance-specific optimal values on corresponding regions. With the test results obtained from HyperPredict$_{\text{clap}}$, we compute the optimal Dice scores across each label. Figure 8 (right) illustrates the varying optimal weights for selected structures. We visualize registration results (left) for two regions; Thalamus and Amygdala, using their optimal values, 0.09 and 0.16 respectively. In
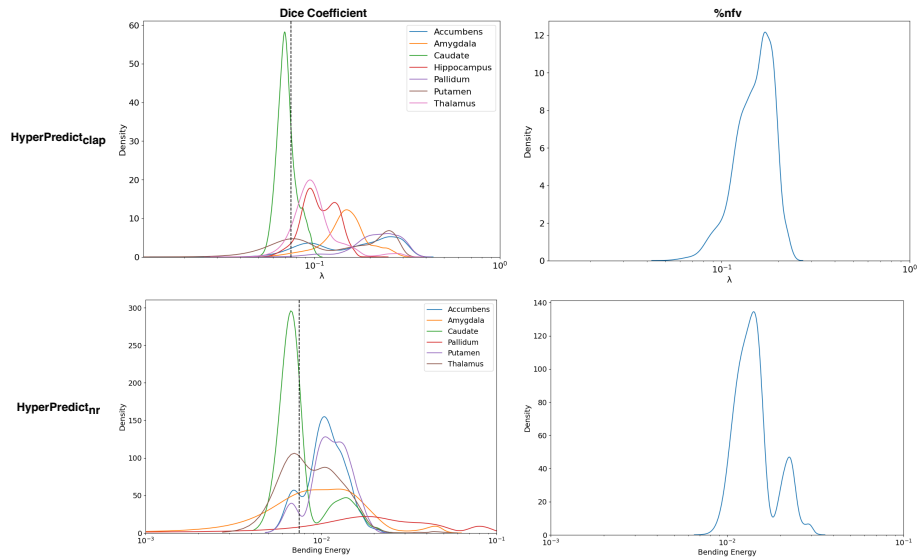
700

Figure 7: Distribution of Optimal Regularization Weights. **Left:** across labels **Right:** across subjects. For HyperPredict$_{\text{clap}}$(top row) and HyperPredict$_{\text{nr}}$(bottom row). The black-dashed lines in the plot represents the optimal value obtained through cross-validation method on both registration.
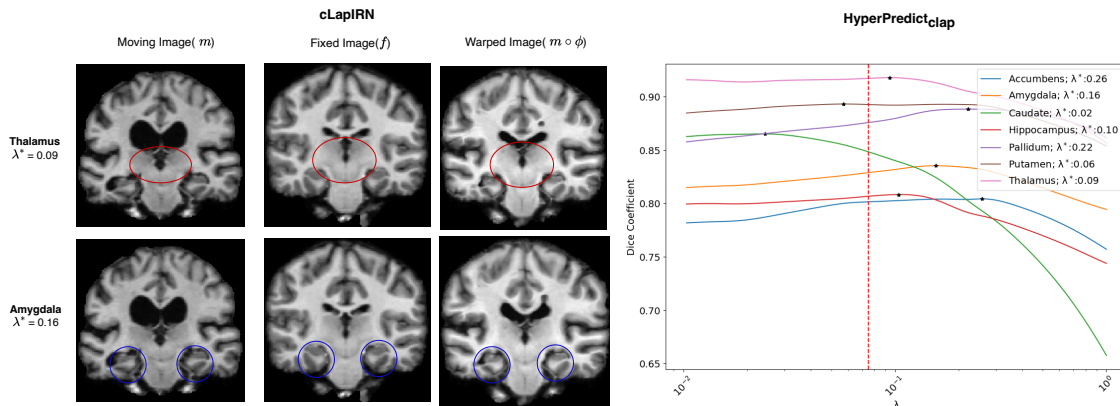

Figure 8: Varying regularization weights per anatomical structure. **Right:** Average Dice scores plotted against sampled regularization weights for selected anatomical regions. **Left:** Registration of an image pair using optimal parameters, with Thalamus and Amygdala as intended regions of interest.

both registrations, we obtain an overall dice score of 0.75 and 0.73 across the entire image. However, when optimizing for Thalamus (using 0.09), the Dice score of Thalamus yields 0.89 and that of Amygdala was 0.80. Similarly, when optimizing for Amygdala (using 0.16), we

obtained 0.78 and 0.80 for Thalamus and Amygdala respectively. Notice we obtain higher Dice scores for the specific region we optimize for, showing the significance of instance-specific regularization. The red-dashed line in the figure represents the optimal value obtained through cross-validation method, 0.075. We present result for HyperPredict$_{nr}$ in Appendix A, Figure 9 (right).

5.4 Computational Efficiency at Test Time

HyperPredict demonstrates notable computational efficiency at test time. A single model is capable of evaluating the effects of hundreds of hyperparameter values without imposing significant computational burden. In Table 1, we present a summary that compares the test time of HyperPredict with other baseline registration methods. It is important to note that while HyperPredict by itself is not a registration method, it acts as a tool that facilitates the selection of optimal hyperparameters. Therefore, the purpose of the comparison is to highlight that unlike HyperPredict, baseline methods are unable to infer the effect of hundreds (or thousands) of hyperparameters within a short period of time - which is a hindrance when trying to select the optimal hyperparameter.

The runtime for methods under our proposed framework (utilizing Symnet encoder) is presented, with testing conducted using 8000 distinct hyperparameter values (HP) and includes the time it takes with and without registration (w/reg). HyperPredict with registration (w/reg) is the time it takes to run all 8000 values and using the optimal for registrating the image pair (either with cLapIRN or Niftyreg method), while HyperPredict without registration (w/o reg) is the time taken to test HyperPredict with 8000 values. We also present results of HyperMorph (Hoopes et al., 2021), highlighted in Section 3 as an additional baseline method, we asterisks it to show that the result is adopted from the paper.

Table 1: Comparison between HyperPredict and other registration methods at test time. Results are presented in seconds.

Method	Test Time(1HP)	Test Time(8000 HP)	
		w/o reg	w/reg
HyperMorph*	2.1	-	16800
Niftyreg	15.0	-	120000
HyperPredict$_{nr}$	-	**0.84**	**15.84**
cLapIRN	0.10	-	800
HyperPredict$_{clap}$	-	**0.84**	**0.94**

6. Discussion

Our experiments demonstrated for the first time that the effects of a regularization hyper-parameters can be predicted for pairwise image registration. We employ two metrics as a means of evaluating the registration; the dice score (for each segmented region), and the number of folded voxels (nfv), however, our framework could be extended to incorporate

additional registration quality metrics.

Accuracy of HyperPredict: In our experiments, we utilize two registration algorithms: cLapIRN and Niftyreg to derive target values for training HyperPredict. Analysis from the residual plots in Experiment 1 reveals that the difference between the predicted and true values are centered around zero. This suggests that on average, our predictive model is making unbiased predictions and does not consistently overestimate or underestimate the predicted metrics. Similarly, the Bland-Altman plots show a good correlation between HyperPredict and cLapIRN registration. With Niftyreg, we find comparable results between the predicted and target dice coefficients, however, our model demonstrates sight limitations in accurately predicting the %nfv. We hypothesize that this effect may be as a result of the use of a more complex similarity function (normalised mutual information) and the algorithmic heuristics, that makes it less predictable than cLapIRN. Another contribution to the sub-optimal predictive performance on %nfv may be the choice of encoder. Currently, we employ two off-the-shelf convolutional encoders. The experiment presented in Appendix C shows that the choice of encoder influences how effective HyperPredict is in estimating the desired metrics, it may be that: (a) the encoded representation itself does not contain sufficient information for HyperPredict$_{nr}$ to infer the %nfv and/or (b) the use of a fixed convolutional encoder limits its ability to capture certain representations of the input specific to the task. In future, it is worth investigating the restrictions associated with encoder choice.

HyperPredict vs Cross-Validation: Cross-validation is the principle existing method for hyperparameter selection. The experiment results in Figure 4 comparing HyperPredict and cross-validation show comparable dice scores using HyperPredict$_{clap}$ and HyperPredict$_{nr}$, with both HyperPredict approaches deriving slightly better results on a number of anatomical regions (Pallidum and Amygdala). We also find that HyperPredict$_{clap}$ produces registration with substantially less voxel folding than cross-validation (Figure 4 top-right). This implies that our approach is able to capture the variability between subjects and identify an improved instance-specific level of regularization.

We perform an additional comparison between both methods by investigating the proportion of examples where HyperPredict over-performs, under-performs, or yields comparable results to cross validation, presented in Figure 5, we term this as "better", "worse" and "equal" cases respectively in the Figure. For both HyperPredict$_{clap}$ and HyperPredict$_{nr}$, the Dice coefficient are comparable with HyperPredict$_{nr}$ having slightly more cases where HyperPredict performs better. HyperPredict$_{clap}$ demonstrates a significant improvement over cross-validation in terms of %nfv, exhibiting lower %nfv values for nearly all subjects (top-right). On the other hand, when comparing HyperPredict$_{nr}$ with cross-validation, there are slightly more instances where cross-validation outperforms HyperPredict$_{nr}$ (bottom-right), the reason for this may be attributed to the limitation in the predictive ability of HyperPredict$_{nr}$.

Finally, the registration of the same image pair using the regularization weight of 0.102 and 0.075 derived from HyperPredict$_{clap}$ and cross-validation shows (in Figure 6) that while we arrive at a comparable dice score of 0.751 and 0.755 respectively, our method performs better in terms of %nfv, with corresponding values of 0.48% and 0.73% respectively. This

shows that utilizing HyperPredict enables us to arrive at more plausible deformation fields.

Distribution of Regularization Weights: Our study finds that different regularization weights yield optimal results depending on the subject and specific anatomical regions of interest. Our proposed framework has the flexibility of selecting such task-specific regularization values at test time as shown in Figures 7 and 8 (right), that can then be used for registration. The effect of such instance specific values is observed in Experiment 3 (Figure 8, left). When we apply the optimal regularization weight of independent anatomical regions (Thalamus and Amygdala) during registration, we derive higher dice scores for that particular region of interest.

We conduct an additional supplementary experiment presented in Appendix A that learns the effect of multiple hyperparameters. We utilize the Niftyreg algorithm due to its ability to incorporate a diverse range of regularization parameters, specifically we learn the effect of two of them simultaneously; bending energy (*be*), and linear elasticity (*le*). We present a heat map in Appendix A, Figure 10 depicting the combined impact of both parameters on the quality metrics across subjects. We observe from the figure that lower values of bending energy and linear elasticity results in our method predicting higher dice scores with more irregularities in the deformation field and vice-versa.

Computational Efficiency: Finally, we analyse the computational efficiency of Hyper-Predict at test time. Exploiting a single trained HyperPredict model at test time means that for any arbitrary image pair, we are able to infer the effect of a range of hyperparameter values from a continuous interval without the need of labelled data while limiting the computational burden. It is important to note that our method is an *amortized inference approach* that incurs an initial computational cost at train time, but results in a computationally efficient prediction at test time. Additionally, without our method a typical approach of selecting an optimal value at test time would involve running multiple independent registrations with different values. This approach becomes impractical as it restricts the search space of values and incurs additional computational costs for each independent registration.

Comparing the run time of a registration model using a single hyperparameter on an image pair to HyperPredict (testing thousands of values on the same image pair) reveals a notable difference in computational efficiency at test time. Table 1 shows that HyperPredict takes less than 0.84 seconds to predict the registration metrics for a single image pair with *8000 different* hyperparameter values. This is approximately 8 times less the time it takes a cLapIRN registration model to derive the results from a *single* hyperparameter value. The first column on the table presents the run time using a single hyperparameter value on various registration methods, the aim of HyperPredict is to test multiple values, hence we don't show results for our method in this column. In the second column, we show the run time with and without registration for 8000 distinct hyperparameter values. With HyperPredict, we can efficiently test and select the best value from all 8000 options. On the other hand, baseline registration methods necessitate more computational resources since the registration using all 8000 values need to be executed and the optimal one chosen based on the registration outcomes. Evidence of this is shown in the last column (w/reg), where it takes approximately 16800 seconds for HyperMorph to run 8000 distinct values on a single

image pair, Niftyreg takes around 120000s (compared to HyperPredict$_{nr}$ that takes 0.84 seconds to run all 8000 values and an additional 15s to run registration on the optimal value). A similar pattern can be observed between cLapIRN and HyperPredict$_{clap}$, with runtime of 800 seconds and 0.94 seconds, respectively. HyperPredict enables a fast and more efficient method of hyperparameter selection at test time based on a flexible criteria, presenting a substantial advantage over baseline parameter selection methods.

7. Limitations and Future Work

The findings from HyperPredict demonstrate the efficacy of our approach in aiding the selection of hyperparameters that align with the unique attributes of the input. One limitation of our approach is the computational cost during training, as it involves running multiple registrations to optimize the MLP. Although, we perform further experiment presented in Appendix E that illustrates that HyperPredict can learn from 25% of the available data while achieving satisfactory result. This significantly decreases the overall registrations that need to be run when utilizing our approach. To circumvent this computational cost at train time and for reproducibility, we ran multiple registrations (for both cLapIRN and Niftyreg) across all image pairs with sampled hyperparameter values. We have made the csv files containing registration results publicly accessible in our GitHub repository.

Our findings indicate that regularization weights vary across different anatomical structures, it is also likely that narrower regions in the brain are more difficult to register, future works will look into learning the importance of different regions and the effect it has on registration. Additionally, we plan to explore other hyperparameters, such as cost functions, as well as incorporating diverse datasets like Alzheimer's and lung data to generalize HyperPredict. Furthermore, experiments in Appendix B and C depicts the influence of model architecture and encoded representations on metric prediction, we intend to explore alternative encoded representations of image pairs. This can include deriving an image representation from a network that predicts the overlap between pairs of images, fine-tuning existing models, or utilizing self-supervised representation learning, we can then analyze the effect any of the methods have on the predicted result. Finally, in computing the number of folded voxels, we considered the entire image - including regions outside the labels, another problem formulation could be to learn the diffeomorphic properties of the image for only regions in the image where anatomical structures exist.

We have made our code, registration results and guidelines available on GitHub and we invite the community to contribute by replicating our experiments on different datasets, conducting further evaluations on new experiments, or expanding the applicability of HyperPredict to different use-cases.

8. Conclusion

The accuracy of non-linear registration algorithms heavily depends on the choice of hyperparameter values, which may vary based on various factors. Thus, selecting a hyperparameter depending on specific use-case is an essential component of image registration. In this study, we propose HyperPredict, an efficient method of learning the effect of hyperparameters on registration, one that eliminates the need of running multiple registrations at test time in

search for an optimal hyperparameter. HyperPredict utilizes an MLP that takes an image pair and desired set of hyperparameter as input and in return, predicts the evaluation metric that corresponds to the input. Our experiments show that by training a single HyperPredict model, we capture the effect of a range of hyperparameter values on an image pair. With this ability, at test time HyperPredict is able to predict the registration results of an image pair with thousands of registration hyperparameter values, giving us the ability to select the optimal value for a specific use-case. In summary, we have presented a novel approach we believe is more efficient in aiding the selection of optimal hyperparameter in image registration. While the experiments presented in this paper is specific to the domain of medical imaging, the general idea of HyperPredict can be adopted in other applications.

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript. We train our models with publicly available data.

Conflicts of Interest

We declare we don't have conflicts of interest.

References

Jesper LR Andersson, Mark Jenkinson, Stephen Smith, et al. Non-linear registration, aka spatial normalisation fmrib technical report tr07ja2. *FMRIB Analysis Group of the University of Oxford*, 2(1):e21, 2007.

John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1): 95–113, 2007.

Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008a.

Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008b.

Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging*, 38(8):1788–1800, 2019.

M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

OASIS Brains. Open access series of imaging studies, 2020.

Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26 (3):297–302, 1945.

Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

Andrew Hoopes, Malte Hoffmann, Bruce Fischl, John Guttag, and Adrian V Dalca. Hyper-morph: Amortized hyperparameter learning for image registration. In *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings 27*, pages 3–17. Springer, 2021.

Sarang Joshi, Brad Davis, Matthieu Jomier, and Guido Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage*, 23:S151–S160, 2004.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Loic Le Folgoc, Herve Delingette, Antonio Criminisi, and Nicholas Ayache. Quantifying registration uncertainty with sparse bayesian modelling. *IEEE transactions on medical imaging*, 36(2):607–617, 2016.

Jun Ma, Michael I Miller, Alain Trouvé, and Laurent Younes. Bayesian template estimation in computational anatomy. *NeuroImage*, 42(1):252–261, 2008.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.

Mingyuan Meng, Lei Bi, Michael Fulham, David Dagan Feng, and Jinman Kim. Enhancing medical image registration via appearance adjustment networks. *NeuroImage*, 259:119444, 2022.

Jonas Mockus. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2:117, 1998.

Marc Modat, David M Cash, Pankaj Daga, Gavin P Winston, John S Duncan, and Sébastien Ourselin. Global image registration using a symmetric block-matching approach. *Journal of medical imaging*, 1(2):024003–024003, 2014.

Tony CW Mok and Albert Chung. Fast symmetric diffeomorphic image registration with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4644–4653, 2020a.

Tony CW Mok and Albert CS Chung. Large deformation diffeomorphic image registration with laplacian pyramid networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pages 211–221. Springer, 2020b.

Tony CW Mok and Albert CS Chung. Conditional deformable image registration with convolutional neural network. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*, pages 35–45. Springer, 2021a.

Tony CW Mok and Albert CS Chung. Conditional deformable image registration with convolutional neural network. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*, pages 35–45. Springer, 2021b.

Marc Niethammer, Roland Kwitt, and Francois-Xavier Vialard. Metric learning for image registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8463–8472, 2019.

Oula Puonti, Juan Eugenio Iglesias, and Koen Van Leemput. Fast and sequence-adaptive whole-brain segmentation using parametric bayesian modeling. *NeuroImage*, 143:235–249, 2016.

Petter Risholm, Firdaus Janoos, Jennifer Pursley, Andriy Fedorov, Clare Tempany, Robert A Cormack, and William M Wells. Selection of optimal hyper-parameters for estimation of uncertainty in mri-trus registration of the prostate. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part III 15*, pages 107–114. Springer, 2012.

Petter Risholm, Firdaus Janoos, Isaiah Norton, Alex J Golby, and William M Wells III. Bayesian characterization of uncertainty in intra-subject non-rigid registration. *Medical image analysis*, 17(5):538–555, 2013.

Torsten Rohlfing. Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable. *IEEE transactions on medical imaging*, 31(2): 153–163, 2011.

Daniel Rueckert, Luke I Sonoda, Carmel Hayes, Derek LG Hill, Martin O Leach, and David J Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.

Jan Rühaak, Thomas Polzin, Stefan Heldmann, Ivor JA Simpson, Heinz Handels, Jan Modersitzki, and Mattias P Heinrich. Estimation of large motion in lung ct by integrating

regularized keypoint correspondences into dense deformable registration. *IEEE transactions on medical imaging*, 36(8):1746–1757, 2017.

Hanna Siebert, Lasse Hansen, and Mattias P Heinrich. Fast 3d registration with accurate optimisation and little learning for learn2reg 2021. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–179. Springer, 2021.

Ivor JA Simpson, Julia A Schnabel, Adrian R Groves, Jesper LR Andersson, and Mark W Woolrich. Probabilistic inference of regularisation in non-rigid registration. *NeuroImage*, 59(3):2438–2451, 2012.

Ivor JA Simpson, Manuel Jorge Cardoso, Marc Modat, David M Cash, Mark W Woolrich, Jesper LR Andersson, Julia A Schnabel, Sébastien Ourselin, Alzheimer's Disease Neuroimaging Initiative, et al. Probabilistic non-linear registration with spatially adaptive regularisation. *Medical image analysis*, 26(1):203–216, 2015.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26. PMLR, 2021.

François-Xavier Vialard and Laurent Risser. Spatially-varying metric learning for diffeomorphic image registration: A variational framework. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part I 17*, pages 227–234. Springer, 2014.

François-Xavier Vialard, Laurent Risser, Darryl D Holm, and Daniel Rueckert. Diffeomorphic atlas estimation using karcher mean and geodesic shooting on volumetric images. In *MIUA*, pages 55–60, 2011.

Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1997.

Jian Wang and Miaomiao Zhang. Bayesian atlas building with hierarchical priors for subject-specific regularization. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*, pages 76–86. Springer, 2021.

Jian Wang, Wei Xing, Robert M Kirby, and Miaomiao Zhang. Data-driven model order reduction for diffeomorphic image registration. In *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, pages 694–705. Springer, 2019.

Miaomiao Zhang, Nikhil Singh, and P Thomas Fletcher. Bayesian estimation of regularization and atlas building in diffeomorphic image registration. In *International conference on information processing in medical imaging*, pages 37–48. Springer, 2013.
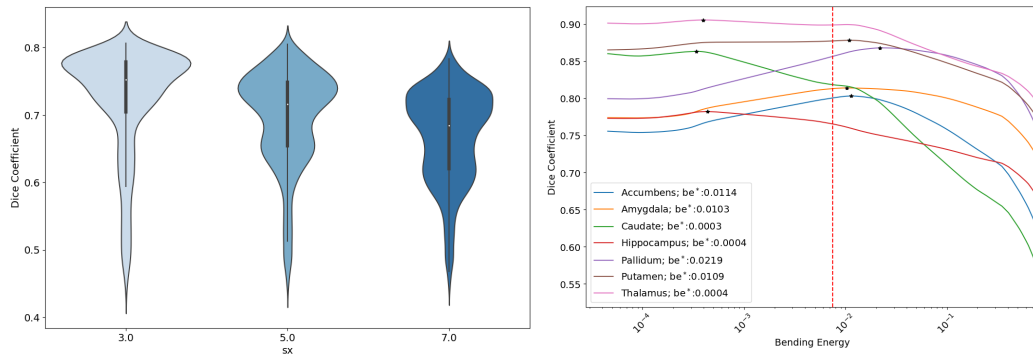
Appendix A. Experiments on Niftyreg

Figure 9: **Left:** Niftyreg yields higher Dice coefficients with smaller spacing values. **Right:** Varying optimal bending energy across selected labels.
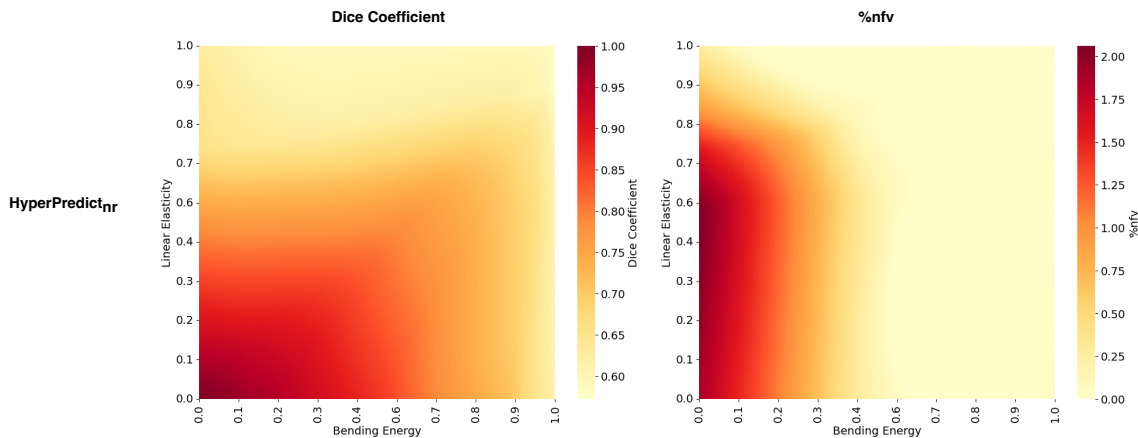

Figure 10: Heatmap depicting the combined effect of both hyperparameters; bending energy and linear elasticity on the dice score and %nfv across subjects.

Figure 10 is a supplementary experiment to investigate the combined effect of multiple regularization on each subject. To conduct this experiment, we train an additional HyperPredict$_{nr}$ model following the methodology outlined in Section 5. However, in this iteration the model specifically focuses on learning the influence of both the bending energy and linear elasticity. Both of which are sampled from a log-normal distribution and serve as input to the MLP at train time. At test time, for both *be* and *le*, we sample 200 log-linearly spaced values between -10 and 0 for each hyperparameter; this means each subject is tested with a combination of 40,000 values. Without a selection criteria, we present a heatmap

above that shows the simultaneous effect of both bending energy and linear elasticity on the Dice coefficient and the folded voxels across subjects.
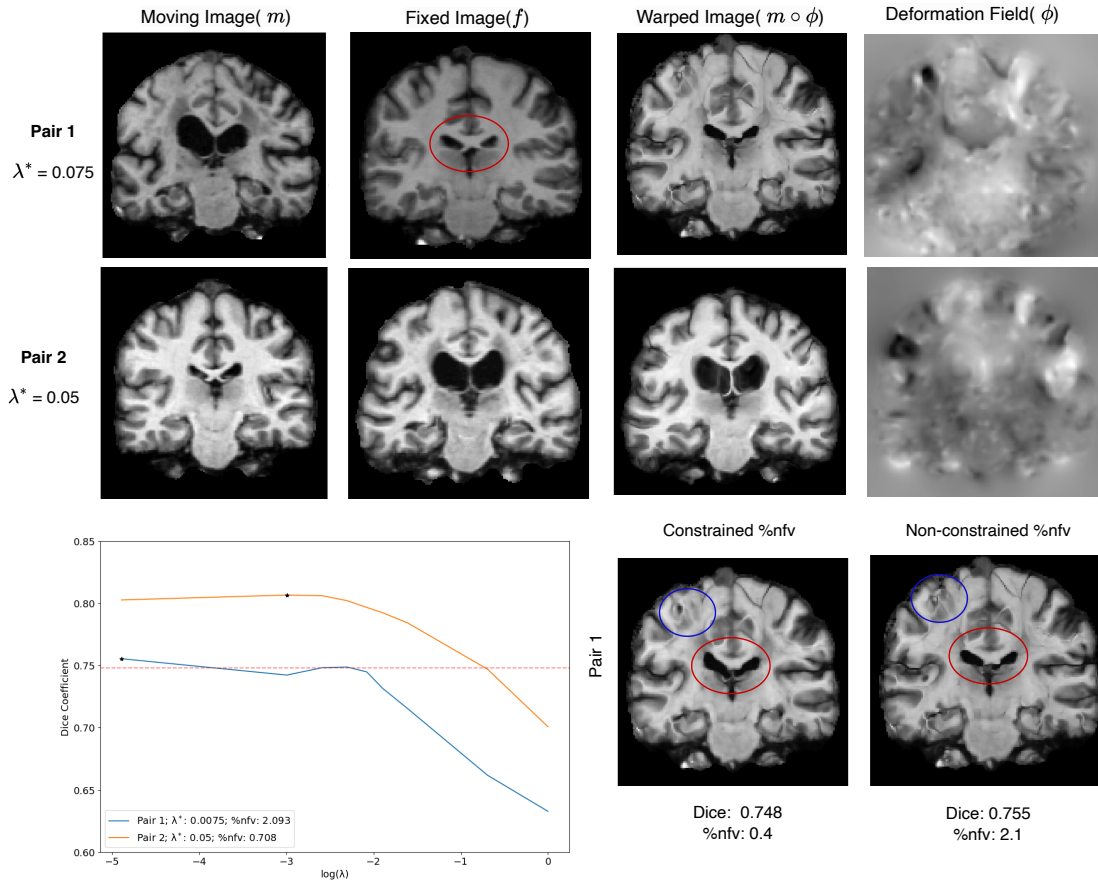

Figure 11: **Applying No Restriction on %nfv**: Since the criteria used to define an optimal value is flexible after training, we change our criteria to consider no restrictions on the %nfv in this experiment, and compare the results with the constrained version. **Bottom Left:** Dice scores plotted against values of λ, the red line is the optimal value inferred with restriction on %nfv for this specific image pair (Pair 1). **Bottom Right:** Comparing warped images with and without restricted %nfv for image Pair 1. The non-constrained warped image (brain image on the right) has a closer similarity to the fixed image compared to the constrained warped image (brain image on the left), this is made evident by the presence of a higher dice score (the red marker depicts visual similarities). However, we expect a less smooth warped image, as highlighted by the increased %nfv from 0.4% to 2.1%, we depict this visually using blue markers.
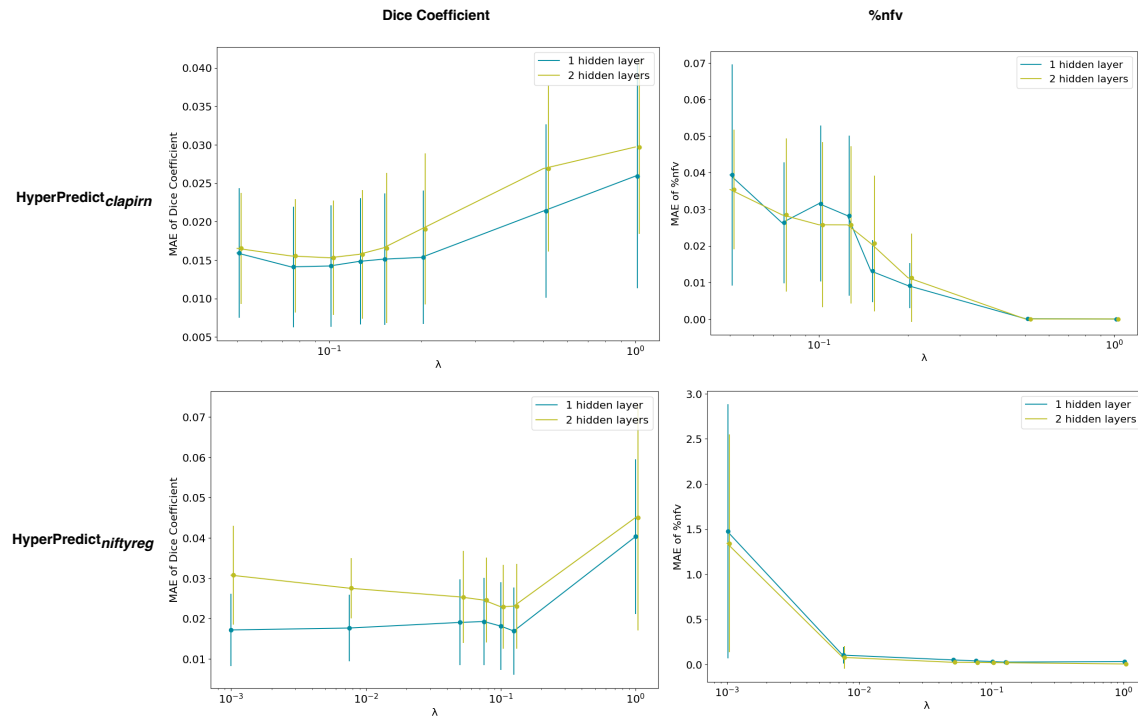
Appendix B. HyperPredict Architecture

Figure 12: Effect of HyperPredict model on two different network architecture (using one and two hidden layers). **Top:** Results using HyperPredict$_{\text{clap}}$. **Bottom:** Results using HyperPredict$_{\text{nr}}$
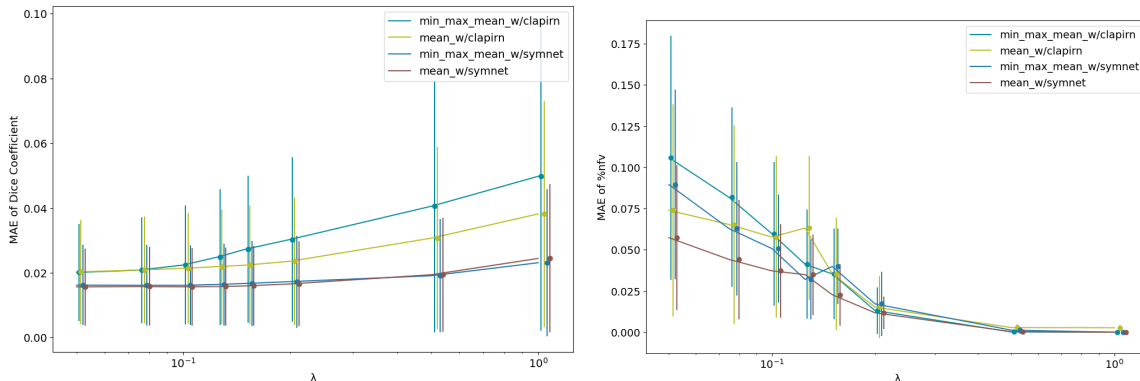
.

Appendix C. Comparing Summary Statistics and Encoder type

Figure 13: Comparing summary statistics and encoder types. Mean absolute error (MAE) of Dice scores (**left**) and %nfv (**right**) using two different summary statistics (taking the *mean* across the channels compared to concatenating the *min*, *max*, and *mean* across the channels) and two different convolutional encoders (cLapIRN and Symnet).

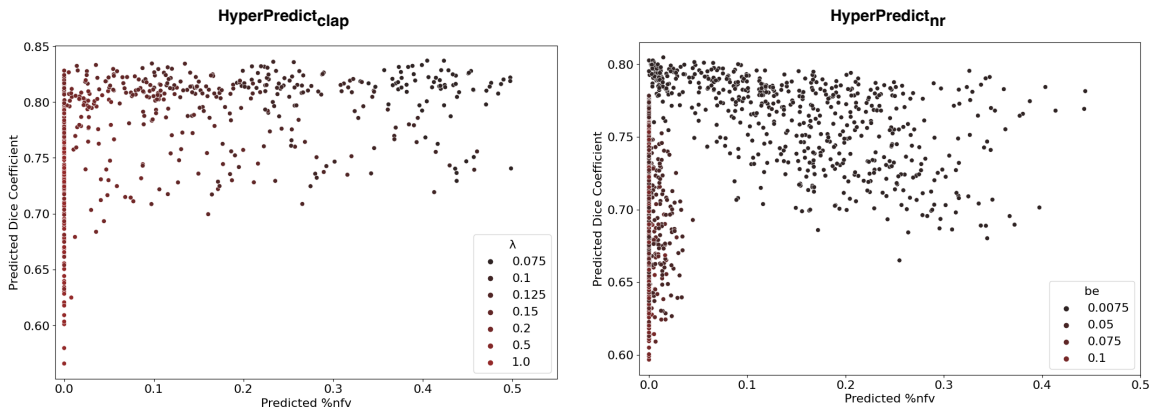## Appendix D. Combined Evaluation of Dice Coefficient and %nfv

Figure 14: Distribution plot of Dice coefficient and %nfv on selected values of λ (left) and bending energy (right), for HyperPredict$_{clap}$ and HyperPredict$_{nr}$ respectively. We show results of a sub-sample for visualization purpose. Both HyperPredict methods predict higher irregularities in the deformation field as the hyperparameter value gets smaller.
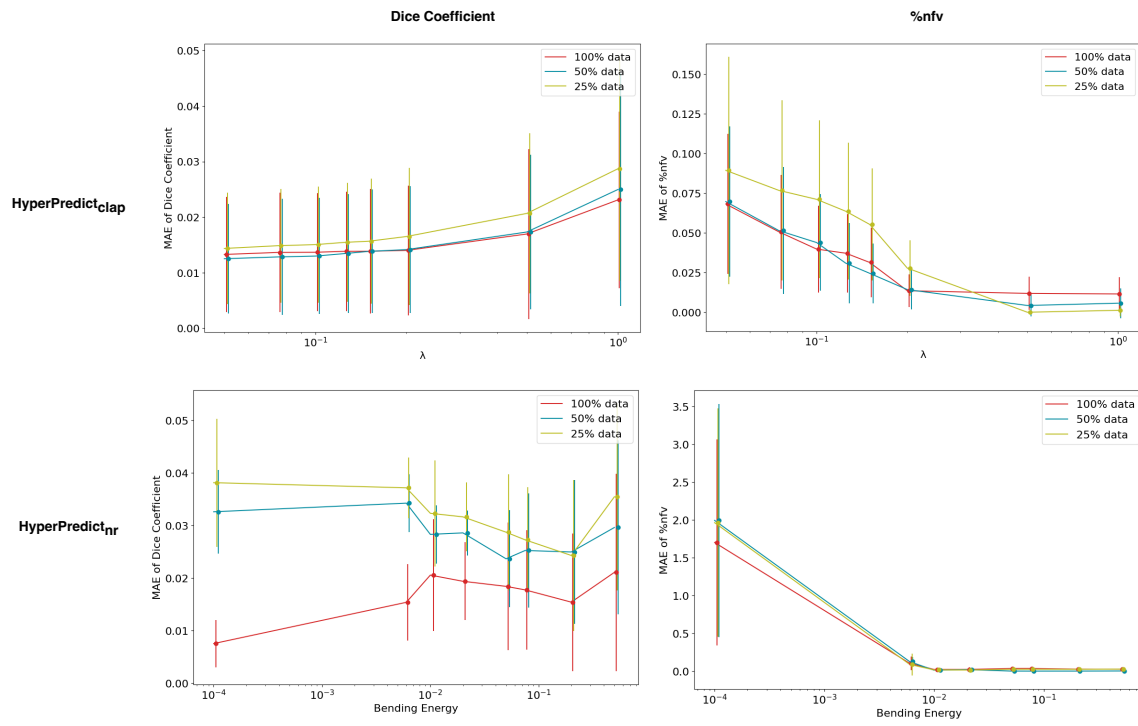
Appendix E. Data Size

Figure 15: Training HyperPredict with different number of training samples (100%, 50% and 25% of image pairs) **Top:** Mean absolute error of dice coefficient and %nfv on HyperPredict$_{clap}$ **Bottom:** Mean absolute error of dice coefficient and %nfv on HyperPredict$_{nr}$
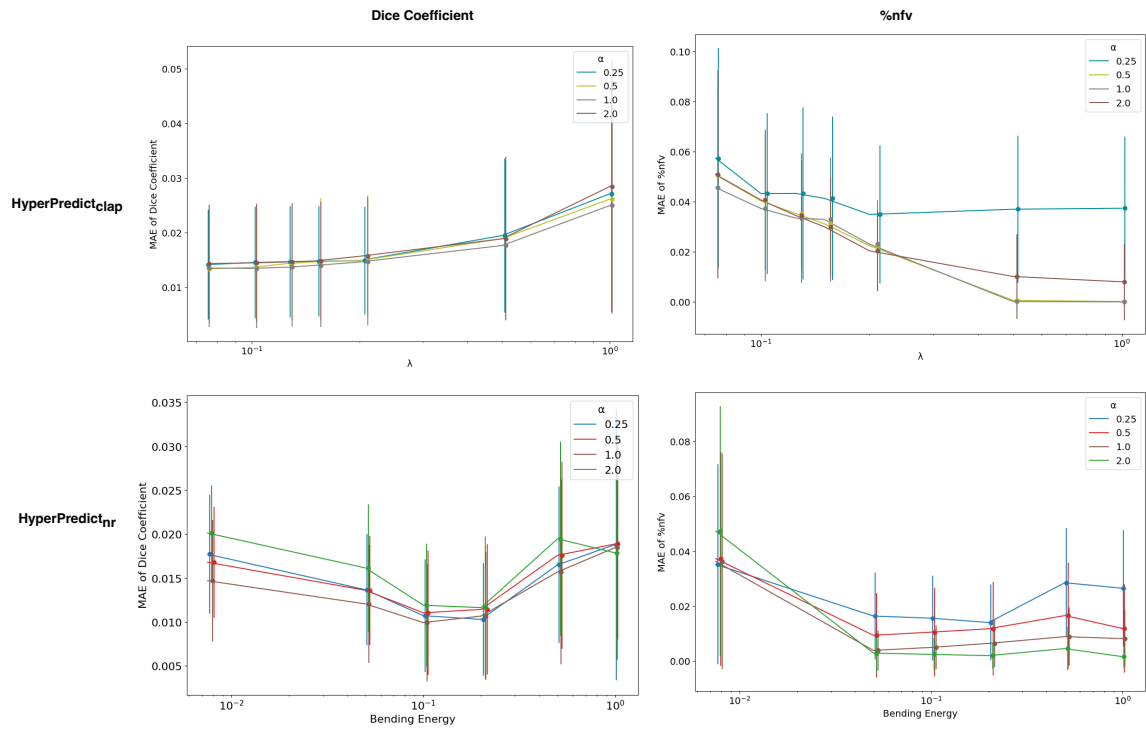
Appendix F. Sensitivity Analysis

Figure 16: Sensitivity analysis result on α using varying values for training. We experimented with smaller values of α, for visualization purposes, we present results for selected four. **Top:** MAE of dice coefficient and %nfv on HyperPredict$_{\text{clap}}$ **Bottom:** MAE of dice coefficient and %nfv on HyperPredict$_{\text{nr}}$