MorphSeek: Fine-grained Latent Representation-Level Policy Optimization for Deformable Image Registration

Supplementary Material

7. Analysis of Latent-Dimension Variance Normalization (LDVN)

We reuse the notation in Sec. 3.3. LDVN modifies the log-likelihood term by introducing a latent-dimension-aware scaling. We define the LDVN-transformed log-likelihood as

$$\hat{\ell}^{(j)} \triangleq \frac{1}{s} \log \tilde{\pi}^{(j)} = \frac{1}{s}\big(\log \pi^{(j)} - \log \bar{\pi}\big), \tag{18}$$

where $s > 0$ is a scaling factor that depends on the latent dimensionality N (specified in Sec. 7.1). The LDVN-based policy loss is then

$$\mathcal{L}_{\text{policy}}^{\text{LDVN}}(\theta_E) = -\frac{1}{J}\sum_{j=1}^{J} A^{(j)}\, \hat{\ell}^{(j)}. \tag{19}$$

Affine invariance under zero-mean advantages. We first show that LDVN does not alter the underlying optimization objective: it only rescales the gradient magnitude while preserving its direction and fixed points.

Consider the more general affine form

$$\hat{\ell}^{(j)} = \alpha \log \pi^{(j)} + \beta \log \bar{\pi} + b, \quad \alpha > 0,\ \beta, b \in \mathbb{R}, \tag{20}$$

and the corresponding policy loss

$$\mathcal{L}_{\text{policy}}^{\text{affine}}(\theta_E) = -\frac{1}{J}\sum_{j=1}^{J} A^{(j)}\, \hat{\ell}^{(j)}. \tag{21}$$

Proposition 1. *Under the zero-mean advantage condition in Eq. 10, the gradient of $\mathcal{L}_{policy}^{affine}$ with respect to the encoder parameters θ_E is*

$$\nabla_{\theta_E} \mathcal{L}_{policy}^{affine} = -\frac{\alpha}{J}\sum_{j=1}^{J} A^{(j)}\, \nabla_{\theta_E} \log \pi^{(j)}.$$

In particular, any affine transform of the form 20 preserves the policy-gradient direction and only rescales its magnitude by the positive constant α.

Proof. Since b does not depend on θ_E, we have

$$\nabla_{\theta_E} \hat{\ell}^{(j)} = \alpha\, \nabla_{\theta_E} \log \pi^{(j)} + \beta\, \nabla_{\theta_E} \log \bar{\pi}.$$

Using the definition of $\log \bar{\pi}$,

$$\nabla_{\theta_E} \log \bar{\pi} = \nabla_{\theta_E} \frac{1}{J}\sum_{k=1}^{J} \log \pi^{(k)} = \frac{1}{J}\sum_{k=1}^{J} \nabla_{\theta_E} \log \pi^{(k)}.$$

Therefore,

$$\begin{aligned}\nabla_{\theta_E} \mathcal{L}_{\text{policy}}^{\text{affine}} &= -\frac{1}{J}\sum_{j=1}^{J} A^{(j)} \Big[\alpha\, \nabla_{\theta_E} \log \pi^{(j)} + \beta\, \nabla_{\theta_E} \log \bar{\pi}\Big] \\ &= -\frac{\alpha}{J}\sum_{j=1}^{J} A^{(j)} \nabla_{\theta_E} \log \pi^{(j)} - \\ &\quad \frac{\beta}{J}\left(\sum_{j=1}^{J} A^{(j)}\right)\left(\frac{1}{J}\sum_{k=1}^{J} \nabla_{\theta_E} \log \pi^{(k)}\right).\end{aligned}$$

By Eq. 10, $\sum_{j=1}^{J} A^{(j)} = 0$, hence the second term vanishes exactly and we obtain

$$\nabla_{\theta_E} \mathcal{L}_{\text{policy}}^{\text{affine}} = -\frac{\alpha}{J}\sum_{j=1}^{J} A^{(j)} \nabla_{\theta_E} \log \pi^{(j)}.$$

Thus the gradient direction coincides with the standard GRPO gradient, up to a global positive scalar α, proving the claim.

Taking $\alpha = 1/s$ and $\beta = -1/s$ recovers LDVN in Eq. 18. Thus LDVN does not change gradient direction or fixed points, and only adjusts the effective update scale.

7.1. Dimension-dependent variance and choice of s

We now analyze how the variance of the log-likelihood grows with the latent dimensionality N and use this to derive a principled choice for the scaling factor s.

For the Gaussian policy in Eq. 8 of the main paper, the log-likelihood of a sampled latent code $\mathbf{z} = (z_1, \ldots, z_N)$ can be written as a sum of N per-dimension contributions:

$$\begin{aligned}&\log \pi(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) \\ &= -\frac{1}{2}\sum_{i=1}^{N}\left[\left(\frac{z_i - \mu_i}{\tau \sigma_i}\right)^2 + \log(2\pi\tau^2\sigma_i^2)\right] \triangleq \sum_{i=1}^{N} X_i,\end{aligned} \tag{22}$$

where X_i denotes the contribution from the i-th latent dimension.

We assume that the per-dimension terms $\{X_i\}$ have uniformly bounded second moments and are at most weakly dependent. Under these mild conditions, the variance of the

sum in Eq. 22 satisfies

$$\mathrm{Var}\Big[\log \pi(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma})\Big] = \mathrm{Var}\Big[\sum_{i=1}^{N} X_i\Big]$$

$$= \sum_{i=1}^{N} \mathrm{Var}(X_i) + 2 \sum_{1 \le i < k \le N} \mathrm{Cov}(X_i, X_k). \quad (23)$$

If $\mathrm{Var}(X_i) \le C$ for all i and the covariance terms are either zero or sufficiently sparse/decaying, the right-hand side grows at most linearly in N, so

$$\mathrm{std}\big[\log \pi(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma})\big] = O(\sqrt{N}). \quad (24)$$

Subtracting the group mean does not change the order of magnitude, so $\mathrm{std}(\log \tilde{\pi}^{(j)}) = O(\sqrt{N})$.

Moreover, for any $s > 0$ and $b \in \mathbb{R}$,

$$\left(\frac{1}{s}\log \pi^{(j)} + b\right) - \overline{\left(\frac{1}{s}\log \pi + b\right)} = \frac{1}{s}\left(\log \pi^{(j)} - \overline{\log \pi}\right). \quad (25)$$

This shows that LDVN is an affine, dimension-aware transformation of the group-relative log-likelihood: it preserves within-group ordering and the policy-gradient direction while only rescaling update magnitude.

Given Eq. 24, we now choose s such that the variance of the LDVN-transformed log-likelihood $\hat{\ell}^{(j)}$ in Eq. 18 remains stable as N grows. Using the fact that scaling a random variable by $1/s$ divides its variance by s^2, we obtain

$$\mathrm{Var}\big[\hat{\ell}^{(j)}\big] = \frac{1}{s^2}\,\mathrm{Var}\big[\log \tilde{\pi}^{(j)}\big] = \frac{1}{s^2}\,O(N). \quad (26)$$

To make this variance $O(1)$, independent of the latent dimensionality, we require

$$\frac{N}{s^2} = O(1) \quad \Longrightarrow \quad s^2 \propto N \quad \Longrightarrow \quad s \propto \sqrt{N}.$$

The above derivation is mathematically analogous to the scaled dot-product attention used in Transformers[49], where the dot product between query and key vectors is divided by $\sqrt{d_k}$ to prevent its variance from growing with the feature dimension d_k. Here, LDVN plays the same role for log-likelihoods in high-dimensional latent spaces: by normalizing $\log \tilde{\pi}^{(j)}$ with $1/\sqrt{N}$, we keep its variance roughly constant across different latent dimensionalities, stabilizing GRPO updates without additional parameters.

In summary, LDVN applies an affine, dimension-aware transformation to the group-relative log-likelihood that (i) preserves policy-gradient direction while rescaling magnitude and (ii) cancels the $O(\sqrt{N})$ growth of its standard deviation.

This turns latent-space policy optimization into a numerically stable procedure even under high-dimensional latent codes, which is crucial for making RL-based registration practically viable beyond low-dimensional rigid transformations.

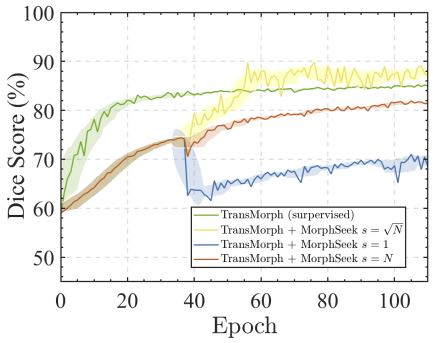

Figure 5. Validation Dice on OASIS for TransMorph under different LDVN scaling factors s.

7.2. Ablation Studies for LDVN

We revisit the Gaussian policy log-likelihood with the LDVN scaling factor s (Eq. 12) and ablate different choices of s. On the TransMorph+OASIS task, we keep all settings identical to the main paper and vary only the LDVN scaling factor,

$$s \in \{1, \sqrt{N}, N\}.$$

We also include a purely supervised TransMorph baseline trained with the Dice loss. Figure 5 reports validation Dice scores over training epochs.

As a result, when $s = N$, the GRPO contribution to the loss is weak; the curve almost coincides with the supervised baseline. When $s = 1$, the variance of $\log \tilde{\pi}^{(j)}$ grows with N, GRPO gradients become noisy, and the model forgets the warm-up representation; the final Dice remains below both the baseline and the other settings. With $s = \sqrt{N}$, the variance is stabilized at $\mathcal{O}(1)$, GRPO updates are stable.

These observations empirically support the choice $s \propto \sqrt{N}$ for high-dimensional latent policies.

7.3. Empirical Check of the Weak-Dependence Assumption

To complement the variance derivation in Sec. 7.1, we empirically examine the weak-dependence assumption on OASIS with the TransMorph backbone ($N \approx 1.6 \times 10^5$). For each checkpoint, we draw 10^3 Monte Carlo latent samples and estimate $\mathrm{Var}(\sum_i X_i)$, where X_i is the per-dimension contribution in Eq. 22.

We report the ratio between empirical variance and the independence baseline ($0.5N$). The observed ratio is 1.01 ± 0.04, indicating negligible cross-dimension correlation in practice. This supports the $O(N)$ variance growth assumption and the choice $s \propto \sqrt{N}$.

7.4. Critical Hyperparameter Sensitivity

We analyze key stability-related hyperparameters on OASIS with TransMorph (50 GRPO epochs). Table 5 sum-

Table 4. Monte Carlo verification of weak dependence on OASIS (TransMorph, 10^3 samples).

Metric	Value
$\mathrm{Var}_{emp}(\sum_i X_i)/(0.5N)$	1.01 ± 0.04

Table 5. Hyperparameter sensitivity on OASIS (TransMorph).

Setting	Dice↑/NJD↓	Observation
Default	88.44/0.06	Stable
$\tau = 1$	83.13/0.04	Under-exploration
$\tau = 15$	55.67/0.01	Exploration collapse
No σ clip	33.05/3.84	Instability
$\sigma_{\max} = 0$	83.87/0.10	Under-exploration
$\lambda_{\mathrm{KL}} = 0.1$	86.20/0.22	Posterior collapse tendency
$\lambda_{\mathrm{KL}} = 0$	49.63/0.00	Collapse
$\omega_{\mathrm{Dice}} = 0$	82.99/0.10	Weaker alignment
$\omega_{\mathrm{NJD}} = 0$	89.52/0.59	Poor regularity

Table 6. Posterior collapse analysis on OASIS. Dice is reported as mean±std (%) over ten latent samples for the same input pair.

Ep. 0 w/o warm-up	Ep. 0 w/ warm-up	Ep. 100 (collapsed)	Ep. 100 (normal)
73.26±0.05	69.00±3.75	88.89±0.00	90.42±0.67

marizes representative failure modes when deviating from the default setting.

In addition, over 20 independent runs, warm-up improves the stable-training success rate from 33% to 79% and reduces the convergence epoch from approximately 120 to 75.

7.5. Posterior Collapse Analysis

We additionally probe posterior collapse by sampling the latent code ten times for the same input at representative checkpoints of VoxelMorph-L on OASIS. Table 6 reports the mean and standard deviation of Dice across the ten samples. Near-zero standard deviation indicates that the encoder has become almost deterministic, removing the exploration signal required by GRPO. This analysis empirically motivates the deterministic warm-up in Eq. 7: initializing the model on the mean code before stochastic sampling reduces collapse risk and helps preserve non-trivial output variance. Without warm-up, or after unstable GRPO under poor hyperparameters, the model can drift toward this regime; the normal warm-started checkpoint instead retains non-trivial output variance.

7.6. Why Keep $\mathcal{L}_{\text{warm}}$ During GRPO?

During GRPO, Dice and NJD act only as proxy rewards. If the optimization is left unconstrained, the policy may exploit these proxies by producing anatomically implausible deformations that still look numerically acceptable. Retaining $\mathcal{L}_{\text{warm}}$ keeps updates close to the anatomy-preserving manifold learned during warm-up.

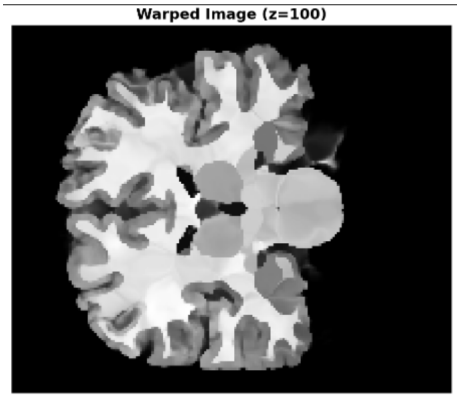

Figure 6. Failure case when removing the similarity term from $\mathcal{L}_{\text{warm}}$ during GRPO. Although proxy metrics can remain deceptively favorable, the warped anatomy becomes smeared and physically implausible, indicating reward hacking.

Figure 6 illustrates this failure mode: without the similarity term in $\mathcal{L}_{\text{warm}}$, GRPO can drive the deformation toward medically meaningless structures that artificially improve overlap-oriented rewards. This observation motivates keeping the warm-up objective during policy optimization, rather than treating it as a pure initialization stage.

Table 7. Additional comparison with multi-stage baselines on OASIS. LapIRN is trained with the same 100 labeled pairs used in our weakly supervised setting.

Method	Dice↑(%)	NJD↓(%)
RIIR (12 steps)	87.76±2.55	0.12±0.02
LapIRN-diff (stage3)	79.70±2.96	0.09±0.03
LapIRN-disp (stage3)	84.52±1.64	3.13±0.41
TransMorph + MorphSeek (3/6)	**88.89±1.82**	**0.06±0.02**

7.7. Additional Comparison with Multi-stage Baselines

Table 7 compares MorphSeek with representative stepwise or cascaded alternatives on OASIS. RIIR is already included in the main paper; we add LapIRN-stage3 here because it is another canonical multi-stage registration baseline. Under the same 100-pair labeled setting, MorphSeek achieves the best overall trade-off between accuracy and deformation regularity.

8. Classical Optimization-Based Baselines

For completeness and to contextualize our learning-based results against strong optimization-based methods, we additionally evaluate two classical non–deep-learning

Table 8. Performance of classical optimization-based baselines on the three benchmarks. We report mean Dice [%] (higher is better), NJD [%] (lower is better), and CPU time per test pair in seconds (lower is better).

Method	Dice [%] ↑			NJD [%] ↓			CPU time [s] ↓		
	OASIS	LiTS	AbMRCT	OASIS	LiTS	AbMRCT	OASIS	LiTS	AbMRCT
SyN	75.53 ± 3.29	79.13 ± 11.26	44.28 ± 28.79	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	48.66 ± 0.00	47.47 ± 0.00	47.01 ± 0.00
deedsBCV	76.38 ± 2.89	77.14 ± 17.77	58.99 ± 20.31	0.23 ± 0.15	0.19 ± 0.12	0.25 ± 0.19	33.18 ± 0.00	31.52 ± 0.02	30.08 ± 0.08

registration algorithms on the same test splits and registration directions as in the main paper. Specifically, we consider SyN from ANTs [1], a classical standard in the field, and deedsBCV [15], a more recent method based on discrete optimization.

For SyN, we follow common practice and use normalized cross-correlation (`syn_metric=CC`) on the mono-modality OASIS and LiTS datasets, and Mattes mutual information (`syn_metric=mattes`) [33] on the cross-modality Abdomen MR←CT (AbMRCT) task. Across all three benchmarks, the multi-resolution schedule is set to `reg_iterations=(60, 40, 20)`, with all remaining parameters kept at their default values.

For deedsBCV, we use self-similarity context (SSC) as the objective function on all datasets. On OASIS, the grid-spacing, search-radius, and quantization-step pyramids are set to $6\times5\times4\times3\times2$, $6\times5\times4\times3\times2$, and $5\times4\times3\times2\times1$, respectively; on LiTS and AbMRCT, the corresponding settings are $8\times7\times6\times5\times4$, $8\times7\times6\times5\times4$, and $5\times4\times3\times2\times1$.

Table 8 summarizes the resulting mean Dice [%], NJD [%], and per-pair CPU time [s] over the test sets. Classical methods remain competitive on OASIS and LiTS but degrade notably on the more challenging AbMRCT task, and they require tens of seconds per case, highlighting the computational overhead of purely optimization-based registration compared with learning-based approaches discussed in the main paper.

9. Discussions: Why MorphSeek Enables Reliable NJD While SPAC Does Not?

A central design choice in MorphSeek is to make the multi-step refinement fully traceable on a *fixed* reference grid. At each step t, MorphSeek maintains an explicit cumulative deformation field Φ_t and updates it by composing the incremental displacement φ_t predicted at that step:

$$\Phi_t = \varphi_t \circ \Phi_{t-1}. \tag{27}$$

Both the loss terms and the warped image $I_m \circ \Phi_t$ are computed using this composed field. Consequently, the final deformation Φ_T is *exactly* the field that produces the reported $I_{\text{warped}} = I_m \circ \Phi_T$, and the NJD metric can be directly evaluated on the same deformation that is responsible for the quantitative results in Table 1. This explicit accumulation makes NJD a well-defined and reproducible measure of deformation regularity for MorphSeek and all refactored baselines.

When we attempted to apply the same NJD protocol to the RL-based SPAC framework, we encountered a structural mismatch between its inference scheme and the requirements for reliable Jacobian analysis. Although SPAC and MorphSeek both adopt multi-step refinement, SPAC does *not* maintain a cumulative deformation field on the original coordinate system. Instead, each predicted single-step displacement is applied directly to the current moving image I_m^t, and only the intermediate images I_m^t and per-step fields ϕ_t are stored. Conceptually, the final warped image can be written as

$$I_{\text{warped}} = I_m \circ \phi_T \circ \phi_{T-1} \circ \cdots \circ \phi_1, \tag{28}$$

where ϕ_t denotes the displacement predicted at step t in the current image coordinates. However, during inference SPAC does not construct or output the exact total deformation Φ_T that maps the original I_m to I_{warped} on a fixed grid.

To make NJD computation possible for SPAC, one must therefore reconstruct a "total" deformation post hoc by composing the saved $\{\phi_t\}_{t=1}^T$ via displacement composition, e.g., using standard ITK-style operators. This inevitably introduces several sources of numerical inconsistency that MorphSeek deliberately avoids by operating on a single reference grid:

- **Interpolation error.** Each resampling of a deformation field smooths the displacement vectors and introduces small geometric deviations; repeating this over many steps amplifies the discrepancy between the reconstructed field and the effective transformation applied during inference.
- **Discretization error.** When the deformation varies rapidly within a voxel, a single sampled displacement cannot faithfully represent the local transformation, leading to biased Jacobian estimates once fields are repeatedly regridded.
- **Non-associativity at the discrete level.** In continuous space, composition is associative, $((\phi_3 \circ \phi_2) \circ \phi_1) = \phi_3 \circ (\phi_2 \circ \phi_1)$. Under "interpolation + grid truncation", different composition orders yield slightly different discrete fields, and these differences accumulate across many refinement steps.

Table 9. Effect of different post-hoc composition schemes on SPAC. Dice and NJD are computed using the reconstructed total deformation $\hat{\Phi}_T$ rather than the original SPAC output (OASIS task).

Composition scheme	Dice (%)	NJD (%)
Original	78.92 ± 5.31	N/A
Vector summation	61.27 ± 7.86	1.42 ± 0.41
Displacement composition + trilinear interp	66.35 ± 4.39	0.25 ± 0.17
Displacement composition + B-spline interp	68.09 ± 6.44	0.30 ± 0.20

In practice, SPAC often uses on the order of $T \approx 20$ refinement steps. After composing $\{\phi_t\}_{t=1}^{T}$ into an approximate total field $\hat{\Phi}_T$ using several reasonable composition schemes, we observe that warping I_m with $\hat{\Phi}_T$ yields segmentations whose Dice scores are more than 10% worse than those obtained from the original SPAC inference I_{warped}. In other words, the reconstructed $\hat{\Phi}_T$ no longer reproduces the reported SPAC behaviour, so any NJD computed on $\hat{\Phi}_T$ would characterize a different, numerically degraded deformation.

Table 9 summarizes this effect on the OASIS task: all post-hoc composition schemes lead to substantial Dice drops and inconsistent NJD values when evaluated on $\hat{\Phi}_T$. These observations indicate that NJD cannot be reliably reported for SPAC without redefining its inference pipeline and output representation. For this reason, we refrain from listing NJD for SPAC in our experiments. In contrast, MorphSeek and the refactored U-Net baselines are expressly designed to maintain an explicit cumulative Φ_T on a fixed grid, ensuring that the reported NJD always reflects the *actual* deformation that produced the corresponding registration results.

10. Implementation Details and Reproducibility

10.1. Loss Definitions and Similarity Metrics

The unsupervised warm-up loss in Eq. 8 of the main paper is

$$\mathcal{L}_{\text{warm}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{sim}}(I_f, I_m \circ \Phi) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\Phi) + \beta_{\text{KL}}\mathcal{L}_{\text{KL}}, \tag{29}$$

where $I_f, I_m : \Omega \to \mathbb{R}$ are fixed and moving images on voxel grid Ω, and $\Phi : \Omega \to \mathbb{R}^3$ is the predicted deformation. We denote the warped image by $\tilde{I}_m = I_m \circ \Phi$ and use a cubic window $\mathcal{N}(\mathbf{p})$ of side length $w = 9$ centered at voxel $\mathbf{p}$ for windowed quantities.

Image similarity. For OASIS and LiTS we use a local MSE similarity:

$$\mathcal{L}_{\text{sim}}^{\text{MSE}}(I_f, I_m \circ \Phi) = \frac{1}{|\Omega|}\sum_{\mathbf{p}\in\Omega}\frac{1}{|\mathcal{N}(\mathbf{p})|}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\big(I_f(\mathbf{q}) - \tilde{I}_m(\mathbf{q})\big)^2. \tag{30}$$

For Abdomen MR←CT we replace MSE with a standard implementation of the MIND descriptor [14], which we use directly as $\mathcal{L}_{\text{sim}}$.

For completeness, we also consider a windowed NCC variant. Let

$$\mu_f(\mathbf{p}) = \frac{1}{|\mathcal{N}(\mathbf{p})|}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} I_f(\mathbf{q}), \tag{31}$$

$$\mu_m(\mathbf{p}) = \frac{1}{|\mathcal{N}(\mathbf{p})|}\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} \tilde{I}_m(\mathbf{q}), \tag{32}$$

and define zero-mean patches $\hat{I}_f(\mathbf{q};\mathbf{p}) = I_f(\mathbf{q}) - \mu_f(\mathbf{p}), \hat{I}_m(\mathbf{q};\mathbf{p}) = \tilde{I}_m(\mathbf{q}) - \mu_m(\mathbf{p})$. The local NCC at $\mathbf{p}$ is

$$\text{NCC}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})}\hat{I}_f(\mathbf{q};\mathbf{p})\hat{I}_m(\mathbf{q};\mathbf{p})}{\sqrt{\sum_{\mathbf{q}}\hat{I}_f(\mathbf{q};\mathbf{p})^2}\sqrt{\sum_{\mathbf{q}}\hat{I}_m(\mathbf{q};\mathbf{p})^2} + \varepsilon}, \tag{33}$$

and the corresponding loss is the negative average correlation:

$$\mathcal{L}_{\text{sim}}^{\text{NCC}}(I_f, I_m \circ \Phi) = -\frac{1}{|\Omega|}\sum_{\mathbf{p}\in\Omega}\text{NCC}(\mathbf{p}). \tag{34}$$

Deformation regularization. Let $\mathbf{u}(\mathbf{x}) = \Phi(\mathbf{x}) - \mathbf{x}$ be the displacement. We use an ℓ_2 diffusion penalty on first-order finite differences:

$$\mathcal{L}_{\text{reg}}(\Phi) = \frac{1}{|\Omega|}\sum_{\mathbf{p}\in\Omega}\sum_{d\in\{x,y,z\}}\big\|\nabla_d\mathbf{u}(\mathbf{p})\big\|_2^2, \tag{35}$$

where ∇_d denotes the discrete difference along spatial direction d.

KL penalty on Gaussian heads. The encoder defines a factorized Gaussian $q_{\boldsymbol{\theta}_E}(\mathbf{z} \mid f_L) = \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$ over the latent tensor $\mathbf{z} = f_L$ with N total dimensions. The KL term is the standard divergence to the unit Gaussian prior:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2N}\sum_{i=1}^{N}\big(\mu_i^2 + \sigma_i^2 - \log\sigma_i^2 - 1\big). \tag{36}$$

10.2. Reward Shaping and Jacobian Regularity

During GRPO fine-tuning we use a reward that combines hard Dice gains with a Jacobian-based regularity term, and an auxiliary soft Dice loss.

Hard Dice for reward shaping. Let $S_f, S_m : \Omega \to \{0, 1, \ldots, K\}$ denote fixed and moving segmentations. For a deformation Φ, we warp the moving labels by nearest-neighbor interpolation,

$$\tilde{S}_m(\mathbf{x}) = S_m(\Phi(\mathbf{x})), \quad \mathbf{x} \in \Omega, \tag{37}$$

and derive one-hot maps $S_f^c, \tilde{S}_m^c : \Omega \to \{0, 1\}$ for each class c. The per-class hard Dice coefficient is

$$\mathrm{Dice}_c^{\mathrm{hard}}(S_f, \tilde{S}_m) = \frac{2 \sum_{\mathbf{x} \in \Omega} S_f^c(\mathbf{x}) \tilde{S}_m^c(\mathbf{x})}{\sum_{\mathbf{x}} S_f^c(\mathbf{x}) + \sum_{\mathbf{x}} \tilde{S}_m^c(\mathbf{x}) + \varepsilon}, \tag{38}$$

and the macro-averaged multi-class Dice is

$$\mathrm{Dice}^{\mathrm{hard}}(S_f, \tilde{S}_m) = \frac{1}{K} \sum_{c=1}^{K} \mathrm{Dice}_c^{\mathrm{hard}}(S_f, \tilde{S}_m). \tag{39}$$

At GRPO step t, each trajectory j produces a deformation $\Phi_t^{(j)}$ and Dice $D_t^{(j)} = \mathrm{Dice}^{\mathrm{hard}}(S_f, S_m \circ \Phi_t^{(j)})$. With baseline deformation Φ_{t-1} and D_{t-1} (identity for $t = 1$), the Dice gain is

$$\Delta D^{(j)} = D_t^{(j)} - D_{t-1}. \tag{40}$$

Jacobian regularity (NJD). For $\Phi(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ we approximate

$$\mathbf{J}_\Phi(\mathbf{x}) = \frac{\partial \Phi(\mathbf{x})}{\partial \mathbf{x}} \approx \mathbf{I}_3 + \nabla \mathbf{u}(\mathbf{x}), \tag{41}$$

and define the set of folding voxels

$$\Omega_-(\Phi) = \big\{\mathbf{x} \in \Omega : \det \mathbf{J}_\Phi(\mathbf{x}) < 0\big\}. \tag{42}$$

The negative-Jacobian determinant percentage is

$$\mathrm{NJD}(\Phi) = \frac{|\Omega_-(\Phi)|}{|\Omega|}, \tag{43}$$

i.e., the fraction of voxels with local foldings.

Step-wise reward and soft Dice loss. The step-wise reward for trajectory j is

$$R^{(j)} = w_{\mathrm{Dice}}\, \Delta D^{(j)} + w_{\mathrm{NJD}}\, \mathrm{NJD}(\Phi^{(j)}), \tag{44}$$

with $w_{\mathrm{Dice}} > 0$ and $w_{\mathrm{NJD}} < 0$. These rewards are group-normalized to compute advantages, which are combined with LDVN-normalized log-probabilities in the policy loss. Compared with optimizing Dice alone, which induces a greedy and deterministic update from the current prediction, GRPO uses relative ranking over sampled trajectories and therefore provides a smoother exploration-based training signal. In practice, evaluating multiple hypotheses per pair helps smooth the highly non-convex registration landscape and makes it easier to escape poor local optima.

Table 10. Approximate latent dimensionality N for each dataset/backbone. All values are computed as $N = H_L W_L D_L C_L$ under the standard input resolutions ($160 \times 192 \times 224$ for OASIS/LiTS and $160 \times 192 \times 192$ for Abdomen MR←CT).

Dataset	VoxelMorph-L	TransMorph	NICE-Trans
OASIS / LiTS	53,760	161,280	53,760
Abdomen MR←CT	46,080	138,240	46,080

In addition, we use a differentiable soft Dice loss. Let $P_c : \Omega \to [0, 1]$ be the warped probabilistic logits for class c after softmax, and $Y_c : \Omega \to \{0, 1\}$ be the one-hot encoding of S_f. The per-class soft Dice is

$$\mathrm{Dice}_c^{\mathrm{soft}}(P, Y) = \frac{2 \sum_{\mathbf{x}} Y_c(\mathbf{x}) P_c(\mathbf{x}) + \varepsilon}{\sum_{\mathbf{x}} Y_c(\mathbf{x})^2 + \sum_{\mathbf{x}} P_c(\mathbf{x})^2 + \varepsilon}, \tag{45}$$

and the multi-class average is

$$\mathrm{Dice}^{\mathrm{soft}}(P, Y) = \frac{1}{K} \sum_{c=1}^{K} \mathrm{Dice}_c^{\mathrm{soft}}(P, Y). \tag{46}$$

The corresponding loss used in Eq. 1 is

$$\mathcal{L}_{\mathrm{Dice}} = 1 - \mathrm{Dice}^{\mathrm{soft}}(P, Y). \tag{47}$$

10.3. Network Architectures and Gaussian Heads

We adopt the official implementations of VoxelMorph-L, TransMorph, and NICE-Trans as backbones and attach a lightweight Gaussian policy head on their top-level encoder features.

VoxelMorph-L. VoxelMorph-L is a symmetric 3D U-Net with encoder channels $[32, 64, 128, 256, 256]$ and decoder channels $[256, 256, 128, 64, 32]$. We take the last encoder feature (before the bottleneck skip connection) as f_L.

TransMorph. For TransMorph, we follow the official 3D large variant, including its encoder–decoder hierarchy and transformer blocks, and use the final encoder feature map as f_L.

NICE-Trans. In NICE-Trans, moving and fixed volumes are encoded independently into 128-channel features, concatenated into a 256-channel tensor, and then fed into the decoder. We place the Gaussian policy head on this concatenated feature map.

Approximate latent dimensionalities N for each dataset–backbone combination are summarized in Table 10.

10.4. Dataset Splits and Pair Construction

We follow Learn2Reg 2021 protocols whenever possible and construct image pairs consistently across backbones. Volumes/scans are first partitioned into disjoint train, validation, and test pools; pair lists are then sampled only within the corresponding pool. Consequently, no test volume appears in warm-up, GRPO, or validation pairs.

OASIS. All volumes are preprocessed and resampled to $160 \times 192 \times 224$. From 414 training volumes we form 400/100/20 pairs for warm-up, GRPO, and validation from the training pool only. The 19 official validation pairs serve as our test set and are never used for training or hyperparameter tuning.

LiTS. LiTS provides 131 contrast-enhanced CT scans with liver and tumor annotations; we use the whole-liver labels only. After preprocessing and resampling to $160 \times 192 \times 224$, we construct 400/100/20/40 pairs for warm-up, GRPO, validation, and test, ensuring that the held-out test pool is fully disjoint from the training and validation pools.

Abdomen MR–CT. This task contains 8 paired MR–CT scans from TCIA and 90 unpaired scans (50 CT from BCV and 40 MR from CHAOS), all resampled to $160 \times 192 \times 192$ with standard intensity preprocessing. From the unpaired scans we form 400/100/20 MR–CT pairs for warm-up, GRPO, and validation, and use the 8 official paired scans as the test set.

For label-efficiency experiments, we subsample the 100 labeled training pairs at different sizes with fixed random seeds. Unless otherwise stated, all refactored backbones share the same pair lists on each benchmark.

10.5. Training Protocols and Hyperparameters

General optimization. Unless noted, all models are trained with Adam, learning rate 10^{-4}, and batch size 1 3D pair. The same learning-rate scale is used for warm-up and GRPO.

Fairness across baselines. VoxelMorph-L, TransMorph, and NICE-Trans are refactored under the same weakly supervised setting and trained on identical pair lists with the same labeled pairs and comparable epoch budgets. CorrMLP, RIIR, SPAC, and WarpDDF+RegCut are reproduced by following their released code or published protocols when a full unification is not directly supported; we report their results under those settings in the main paper.

Gaussian head constraints. The two $1 \times 1 \times 1$ convolutional heads that output $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}$ are regularized as in Eqs. 3–4 of the main paper. A representative setting (TransMorph on OASIS) uses $\lambda_{\text{scale}} = 10$, $\sigma_{\min} = -10$, $\sigma_{\max} = 3$.

Table 11. Efficiency analysis on OASIS. MorphSeek adds less than 3% parameters and near-linear runtime growth with refinement steps.

Baseline	Model Parameters			GPU Inference Time (ms)	
	Original	+Δ Abs	+Δ Rel	Original	+MorphSeek 1/2/3 step(s)
VoxelMorph-L	27.05M	+0.13M	+0.48%	625	685 / 1387 / 2022
TransMorph	46.77M	+1.18M	+2.53%	401	444 / 900 / 1376
NICE-Trans	5.71M	+0.13M	+2.27%	406	431 / 864 / 1295

Warm-up stage. Warm-up optimizes the loss in Sec. 10.1. For TransMorph on OASIS we use $\lambda_{\text{reg}} = 1$ and $\beta_{\text{KL}} = 10^{-4}$; other dataset–backbone combinations choose values of the same order. We run warm-up such that each training pair is seen roughly ten times (about $O(50)$ epochs under our standard settings) and select the checkpoint with the best validation Dice for subsequent GRPO.

GRPO stage. GRPO uses the latent-space policy in Sec. 3.3 with the reward in Sec. 10.2. In our main configuration (e.g., TransMorph on OASIS), we use $J = \text{Trajs} = 6$ trajectories per state and $T = \text{Steps} = 3$ refinement steps. Typical reward weights are $w_{\text{Dice}} = 10$ and $w_{\text{NJD}} = -100$.

The overall GRPO loss is

$$\mathcal{L}_{\text{GRPO}} = \mathcal{L}_{\text{policy}} + \lambda_{\text{warm}} \mathcal{L}_{\text{warm}} + \lambda_{\text{Dice}} \mathcal{L}_{\text{Dice}}, \tag{48}$$

with $\lambda_{\text{warm}} = 0.8$ and $\lambda_{\text{Dice}} = 0.2$ in all main experiments. The exploration temperature τ is linearly annealed from $\tau_{\text{init}} = 10$ to $\tau_{\min} = 2$ (e.g., decreasing by 1 every 10 epochs). GRPO typically traverses the training set on the order of 30–60 epochs; final models are selected by the best validation Dice.

10.6. Hardware and Software Environment

Experiments are conducted on a Linux cluster with up to eight NVIDIA A800-SXM4-80GB GPUs and dual Intel Xeon Silver 4316 CPUs per node. We use Ubuntu 20.04.6 LTS, Python 3.x, and PyTorch 2.3.0. Medical image I/O and preprocessing rely on SimpleITK 2.5.2 together with standard NumPy and SciPy utilities.

10.7. Efficiency Analysis

Table 11 reports the structural and runtime overhead introduced by the RL-friendly refactoring on OASIS. Across all three backbones, MorphSeek adds less than 3% parameters, keeps single-step inference close to the original models, and exhibits near-linear latency growth with the number of refinement steps.

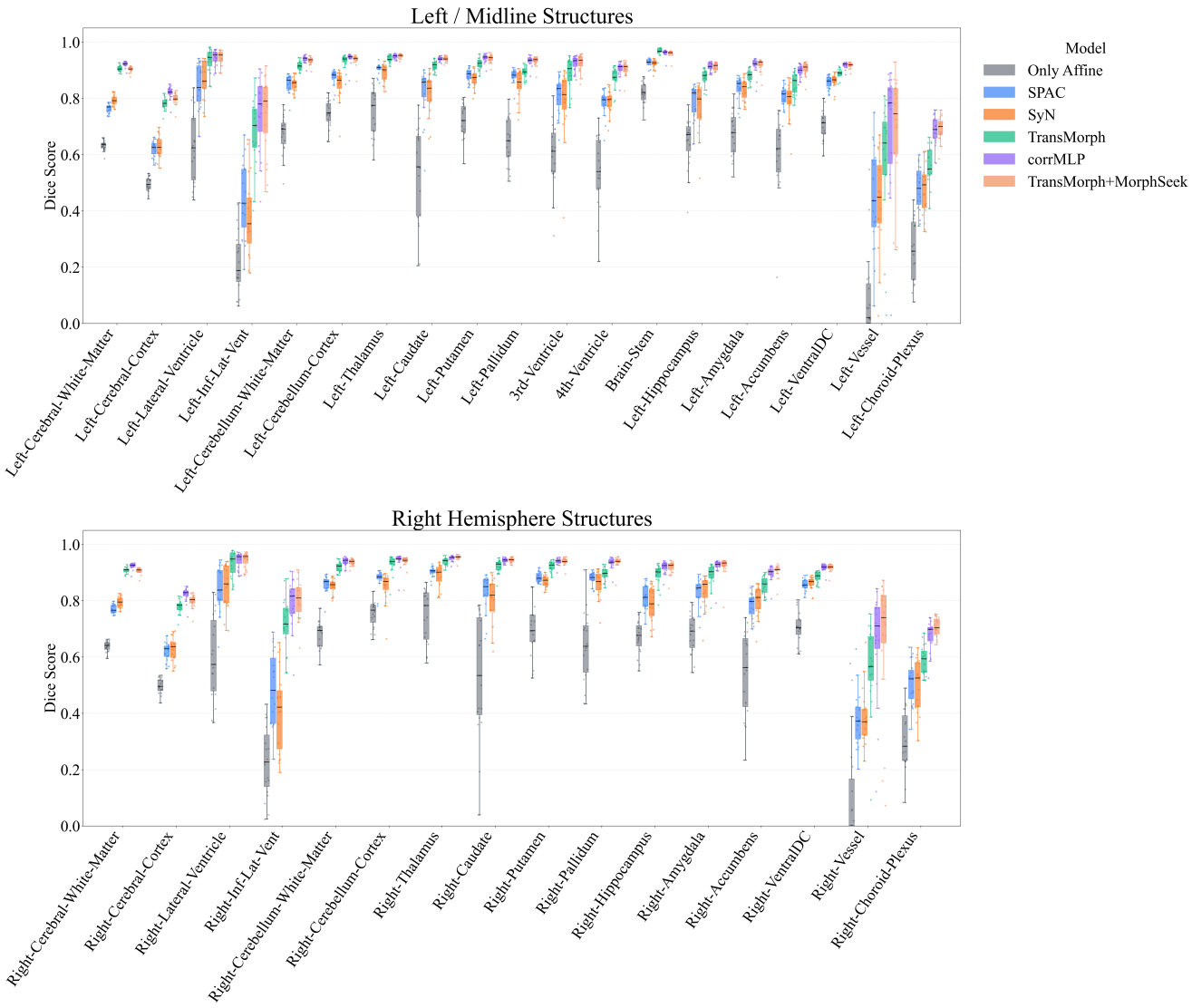

Figure 7. Label-wise Dice on OASIS (SPAC: Steps = 20, TransMorph+MorphSeek: Steps/Trajs = 3/6)

11. Additional Visual Results

To complement the quantitative results in the main paper, we provide two additional visualizations on the OASIS brain MRI benchmark. Figure 7 reports label-wise Dice distributions on the test set for SyN, SPAC, CorrMLP, TransMorph, and TransMorph+MorphSeek. Figure 8 shows a representative registration example, comparing the fixed and moving images with the warped outputs of these methods in three orthogonal views.

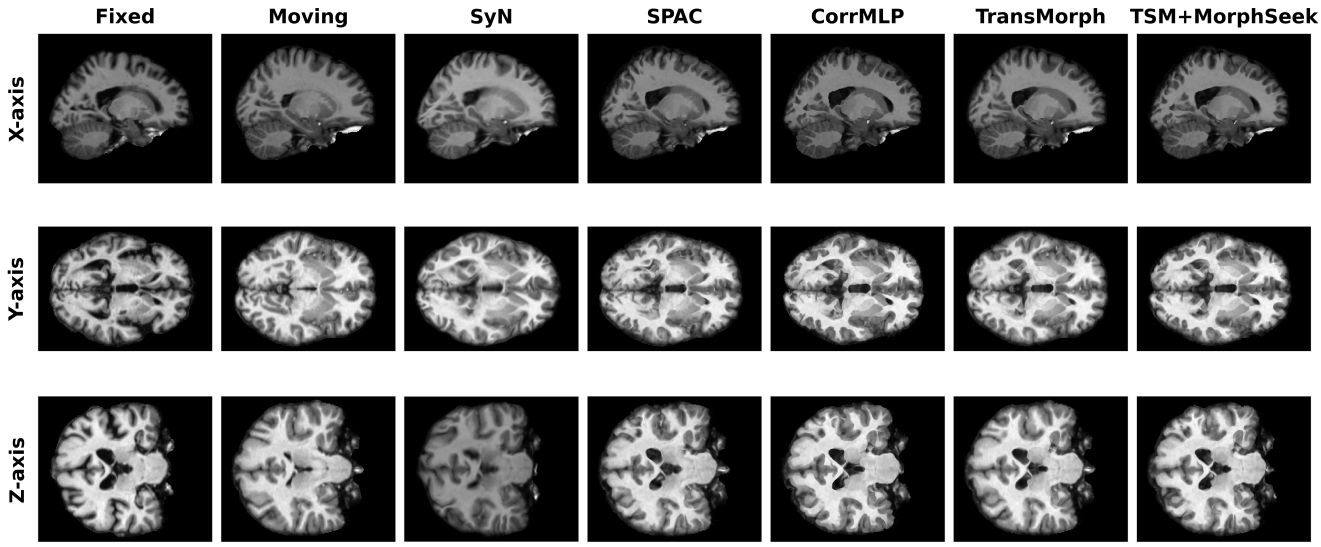

Figure 8. A Qualitative Registration Example on OASIS (SPAC: Steps = 20, TransMorph+MorphSeek: Steps/Trajs = 3/6)