TEXT-ASSISTED UNSUPERVISED DOMAIN ADAPTATION FOR IMAGE REGRESSION THROUGH SEMANTIC RE-ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised domain adaptation for image regression transfers continuous prediction rules without target labels. Existing approaches align visual representations, but feature alignment alone need not preserve the numerical relationships required for regression. Pretrained text encoders offer an alternative representation of numerical attributes, yet target text must be constructed from uncertain predictions. We first analyze the reliability tradeoff of this representation. Deterministic target text adds no information conditional on its complete visual input, but can benefit a restricted regressor through re-encoding. A target absolute-risk bound separates source fitting, domain discrepancy, predictor compatibility, and text-construction error; squared-loss analysis further characterizes branch agreement and conditional error propagation. These results motivate retaining visual information, attending to pseudo-label reliability, and jointly supervising modality-specific and fusion predictions. Guided by these insights, we propose TCC, which encodes source labels and target pseudo-labels, trains three regression branches with a residual-informed objective, and updates the target text by fusing predictions with the preceding text. Experiments on dSprites, MPI3D, and BIWI evaluate the framework through baseline comparisons, branch ablations, and representation and parameter diagnostics. Our source code will be made publicly available.

1 INTRODUCTION

Unsupervised domain adaptation for image regression transfers a continuous prediction rule from labeled source images to an unlabeled target domain. Applications include depth estimation (Godard et al., 2016) and head-pose prediction (Ruiz et al., 2018). Unlike category prediction, regression must preserve numerical relationships under changes in appearance and acquisition conditions (Chen et al., 2021). Aligning feature distributions can reduce domain mismatch without ensuring that the aligned features support the same regression rule.

Existing approaches address this problem through subspace alignment (Chen et al., 2021), inverse Gram alignment (Nejjar et al., 2023), conditional invariant representations (Yang et al., 2024), and uncertainty-guided alignment (Nejjar et al., 2025). These strategies motivate a complementary question: can a pretrained text representation of numerical attributes help preserve a useful regression rule across domains? Unlike fixed-category descriptions, the target text for continuous regression must be constructed from estimates. Understanding its value, therefore, requires analyzing both the representation it provides and the errors introduced by its construction.

We first establish a theoretical account of text-assisted regression (Section 3). Deterministic text re-encoding cannot improve Bayes MSE given the complete visual feature, but it can change the functions accessible to a restricted regressor. We then derive a target absolute-risk bound separating source fitting, reference-input discrepancy, predictor compatibility, and pseudo-label-induced text error. This bound explains why source-side label relevance alone is insufficient: a text decoder can fit source labels while reproducing erroneous target pseudo-labels. A squared-loss decomposition identifies implicit agreement among supervised branches, while a conditional refinement bound exposes how prediction and text sensitivity affect error propagation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These findings lead to two design challenges. Text should provide a useful encoding without replacing the visual evidence from which target estimates originate. At the same time, training must combine modality-specific predictions while accounting for the possibility that incorrect pseudo-labels propagate through text updates. Agreement alone does not ensure accuracy, and refinement requires more than repeatedly decoding the previous prediction.

Guided by these insights, we propose TCC (Section 4). Source labels and target pseudo-labels are encoded as task-related text. A visual and a textual auxiliary regressor supervise their respective projections, while a fusion regressor combines both representations; we call these roles competition and cooperation. Prototype similarity selects initial target samples, and subsequent fusion predictions use the preceding text to update pseudo-labels and text. The branches use the Residual-Informed Loss (RI-Loss) of Wang et al. (2026). The analysis motivates these choices without guaranteeing similarity calibration, error contraction, or the benefit of the additional RI term.

Our contributions follow this theory-to-method progression: (1) an analysis of generated text's information boundary, target risk, branch agreement, and conditional refinement; (2) a text-assisted adaptation framework that combines visual–text fusion, auxiliary branch supervision, and iterative pseudo-label re-encoding; and (3) evaluation on dSprites, MPI3D, and BIWI through baseline comparisons, branch ablations, and representation and parameter diagnostics. Detailed proofs and experimental analyses are provided in the appendix.

2 Related Work

2.1 Domain adaptation regression

General adaptation methods reduce differences between source and target representations through distribution matching (Long et al., 2015), adversarial domain prediction (Ganin et al., 2016), optimal transport (Courty et al., 2017), and predictor discrepancy (Saito et al., 2018). These approaches provide useful tools for transferring predictors, but continuous outputs require attention to the numerical relationships retained after alignment. Marginal feature similarity alone does not determine whether a common regressor fits both domains.

Regression-oriented methods address this structure more directly. RSD aligns representation subspaces (Chen et al., 2021), while DARE-GRAM aligns inverse Gram matrices (Nejjar et al., 2023). COD studies conditional invariant representations (Yang et al., 2024), and UGA uses uncertainty to guide alignment (Nejjar et al., 2025). Other directions include contrastive learning (Wang et al., 2022), multi-source adaptation (Ding et al., 2022), and causal structure (Tsai et al., 2024). Together, these methods motivate evaluating both representation geometry and the prediction rule supported by that geometry.

TCC augments visual features with pretrained encodings of labels or pseudo-labels. Source text uses a true label, whereas target text uses a prediction. Our analysis distinguishes these input laws and separately retains predictor compatibility; re-encoding alone need not remove domain shift.

2.2 Vision-language prediction and generated text

CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) learn shared image–text representations from large-scale pretraining. Adaptation methods use textual guidance for distillation (Li et al., 2023) and category-specific prediction (Bai et al., 2024). Category descriptions can provide a fixed set of semantic anchors. In continuous regression, the numerical content of a prompt must instead be constructed from labels or estimates, and its precision can matter to prediction.

A pretrained text encoder need not preserve numerical distances: nearby values may tokenize differently. We therefore measure pseudo-label-induced embedding error directly. Numerical Lipschitz continuity is assumed only for the conditional refinement result, not for the main adaptation bound.

Unlike an independently observed description, generated target text is deterministic given the complete set of visual features and fixed mappings. Its potential benefit lies in the approximation provided by the available regressor, thereby motivating the retention of a visual pathway alongside text-conditioned fusion.

3 Analysis of Text Re-encoding for Domain Adaptation

We distinguish the information in generated text from its representation effects, then bound target risk when true-label text is replaced by pseudo-label text. Appendix A supplies proofs, notation, refinement analysis, and the branch-loss decomposition.

3.1 Setup and information available at prediction time

Let P_S and $P_{\mathcal{T}}$ be the source and target joint distributions of (X, Y), where $Y \in \mathbb{R}^d$. Write $V = \phi(X)$ and $e(y) = \psi(\mathcal{P}(y))$ for the visual representation and the prompt-encoding map. Let q denote a fixed pseudo-label predictor, including the text-prototype initialization or the preceding refinement steps. Source training uses $T_S = e(Y_S)$, whereas target prediction uses $\widetilde{T} = e(q(V))$. The reference feature $T^* = e(Y)$ is available on the source domain and is an analytical device on the target domain; target labels are never used to construct inference inputs.

All results fix the encoders, the prompt/prototype bank, and the fitted predictors, and concern population risks. Different source and target labeling rules are allowed; their compatibility enters the risk bound.

Proposition 1 (Bayes risk and deterministic text re-encoding)**.** *Assume $\mathbb{E}\|Y\|_2^2 < \infty$. For an input W, define $\mathcal{R}_2^*(W) = \inf_g \mathbb{E}\|Y - g(W)\|_2^2/d$, where the infimum ranges over all measurable predictors. For any additional feature T,*

$$\mathcal{R}_2^*(V) - \mathcal{R}_2^*(V, T) = \frac{1}{d}\mathbb{E}\big\|\mathbb{E}[Y \mid V, T] - \mathbb{E}[Y \mid V]\big\|_2^2. \tag{1}$$

If $T = e(q(V))$ is a deterministic measurable function of V, then

$$I(T; Y \mid V) = 0, \qquad \mathcal{R}_2^*(V, T) = \mathcal{R}_2^*(V). \tag{2}$$

Thus, the generated text cannot reduce the Bayes MSE given the complete V. It can still help a restricted predictor or supplement a compressed projection $U = a(V)$: Appendix A.2 gives an exact example. A lower conditional-entropy bound alone does not imply improved MSE (Appendix A.6).

3.2 Target risk with pseudo-label-derived text

For analysis, use coordinate-averaged absolute loss: $\ell_1(u, y) = \|u - y\|_1/d$. For a class $\mathcal{H}$ of predictors on $z = (v, t)$, define

$$\begin{aligned} R_D^*(f) &= \mathbb{E}_D\, \ell_1(f(V, T^*), Y), \qquad D \in \{S, \mathcal{T}\}, \\ \widetilde{R}_{\mathcal{T}}(f, q) &= \mathbb{E}_{\mathcal{T}}\, \ell_1(f(V, e(q(V))), Y). \end{aligned} \tag{3}$$

Here the superscript $*$ denotes *reference text*, not a Bayes optimum. Let Q_D^* be the distribution of $Z^* = (V, e(Y))$ in domain D. Following the loss-specific discrepancy framework of Mansour et al. (2009), set

$$\mathrm{disc}_{\mathcal{H}}(Q_S^*, Q_{\mathcal{T}}^*) = \sup_{f,g \in \mathcal{H}} \left|\mathbb{E}_{Q_S^*}\ell_1(f(Z^*), g(Z^*)) - \mathbb{E}_{Q_{\mathcal{T}}^*}\ell_1(f(Z^*), g(Z^*))\right|, \tag{4}$$

$$\Lambda_{\mathcal{H}} = \inf_{g \in \mathcal{H}} \{R_S^*(g) + R_{\mathcal{T}}^*(g)\}, \qquad \Delta_{\mathcal{T}}(q) = \mathbb{E}_{\mathcal{T}}\|e(q(V)) - e(Y)\|_2. \tag{5}$$

Theorem 2 (Adaptation risk with text-construction error)**.** *Suppose the risks and discrepancy above are finite. Let $f \in \mathcal{H}$ satisfy, for the relevant visual and text inputs,*

$$\frac{1}{d}\|f(v, t) - f(v, t')\|_1 \le L_f\|t - t'\|_2. \tag{6}$$

Then

$$\boxed{\widetilde{R}_{\mathcal{T}}(f, q) \le R_S^*(f) + \mathrm{disc}_{\mathcal{H}}(Q_S^*, Q_{\mathcal{T}}^*) + \Lambda_{\mathcal{H}} + L_f\Delta_{\mathcal{T}}(q).} \tag{7}$$

The four terms distinguish source fitting, reference-input discrepancy, cross-domain predictor compatibility, and pseudo-label-induced text error. In particular, a source text decoder can fit R_S^* while reproducing target pseudo-label errors. The bound measures embedding error directly: tokenization

and prototype matching need not be smooth in numerical labels. Finite-weight affine/ReLU fusion heads admit a finite L_f, but the architecture does not ensure that it is small.

This is a population comparison, not a computable training objective: $Q^*_{\mathcal{T}}$ and $\Lambda_{\mathcal{H}}$ involve unobserved target labels. Appendix A.3 gives the proof and statistical scope. The following results connect text-construction error and branch supervision to the squared training loss.

3.3 Squared-error sensitivity and branch agreement

For connection to the MSE training loss, define $R^*_{D,2}$ and $\widetilde{R}_{\mathcal{T},2}$ by replacing ℓ_1 with $\ell_2(u,y) = \|u-y\|_2^2/d$. If $\|f(v,t)-f(v,t')\|_2/\sqrt{d} \le L_{f,2}\|t-t'\|_2$, Minkowski's inequality gives

$$\sqrt{\widetilde{R}_{\mathcal{T},2}(f,q)} \le \sqrt{R^*_{\mathcal{T},2}(f)} + L_{f,2}\sqrt{\mathbb{E}_{\mathcal{T}}\|e(q(V)) - e(Y)\|_2^2}. \tag{8}$$

The embedding-error term separates the quality of the numerical estimate from the sensitivity of its text encoding. Appendix A.4 proves this inequality and derives a conditional refinement bound; repeated updates need not improve pseudo-label accuracy monotonically.

Let $p_V, p_T, p_F \in \mathbb{R}^d$ be the three branch predictions on one sample, and let $\bar{Y}$ be its available training target: the true source label or the current target pseudo-label. Define $w_V = \lambda_1$, $w_T = \lambda_2$, and $w_F = \lambda_3$, all nonnegative with positive sum, and $a_j = w_j/(w_V + w_T + w_F)$ for $j \in \{V, T, F\}$.

Proposition 3 (Decomposition of branch-wise squared loss)**.** *Writing $\bar{p} = \sum_j a_j p_j$, we have*

$$\sum_j a_j \ell_2(p_j, \bar{Y}) = \ell_2(\bar{p}, \bar{Y}) + \frac{1}{d}\sum_j a_j \|p_j - \bar{p}\|_2^2. \tag{9}$$

Thus, the squared-error component combines fitting the available target with an implicit penalty on prediction disagreement. The visual and textual auxiliary heads provide direct supervision to their respective projections, while the fusion head is trained on both. The average $\bar{p}$ is used only in this identity; inference still uses the fusion head. The result supports an agreement interpretation, but neither enforces representation diversity nor guarantees correct predictions when all branches share an erroneous pseudo-label. It also does not decompose the nonlinear HSIC term in RI-Loss.

3.4 Conditions for controlling refinement error

To study refinement, let q_0 be prototype initialization and let $q_c(V) = f_F^{(c)}([V; e(q_{c-1}(V))])$ be the prediction obtained after training cycle c. Define

$$\varepsilon_c = \left(\frac{1}{d}\mathbb{E}_{\mathcal{T}}\|q_c(V) - Y\|_2^2\right)^{1/2}, \qquad b_c = \sqrt{R^*_{\mathcal{T},2}(f_F^{(c)})}.$$

Corollary 4 (Conditional propagation through refinement)**.** *Suppose $f_F^{(c)}$ has text sensitivity $L_{c,2}$ as above and, on target pairs $(y, y') = (q_{c-1}(V), Y)$ almost surely,*

$$\|e(y) - e(y')\|_2 \le L_e\|y - y'\|_2/\sqrt{d}.$$

Then $\varepsilon_c \le b_c + L_{c,2}L_e\varepsilon_{c-1}$. If $b_c \le b$ and $L_{c,2}L_e \le \kappa < 1$ for all $c = 1, \dots, C$, then

$$\varepsilon_C \le \kappa^C\varepsilon_0 + b\frac{1-\kappa^C}{1-\kappa}. \tag{10}$$

The proof and an exact error-change identity are in Appendix A.4. The bound describes sufficient sensitivity and reference-risk conditions; it does not assert that the learned models satisfy them. In particular, a text branch that only decodes the preceding pseudo-label need not correct its error. Even when $\kappa < 1$, the bound controls an error level and does not imply monotonic improvement in every realized cycle.

Implications for method design. The analysis motivates three choices. First, Proposition 1 locates the potential benefit in representation rather than new target evidence. We therefore retain visual features and fuse them with pretrained text embeddings rather than relying on text alone. Second, the text-error term in Theorem 2 and the propagation bound in Corollary 4 motivate attention to initialization and update quality. Then we select initial target samples based on prototype similarity and refresh pseudo-labels via fusion with the preceding text. These are practical heuristics, not guarantees of lower embedding error or contraction. Third, Proposition 3 motivates jointly supervising visual, text, and fusion predictions to combine target fitting with implicit agreement. The next section implements these choices; the analysis neither uniquely determines the architecture nor guarantees the benefit of the additional RI regularizer.

4 METHOD

Following these design implications, TCC combines text construction and fusion-based refinement (Section 4.1) with joint branch supervision (Section 4.2). It learns from labeled source data $\mathcal{D}_S = \{(X_S^i, Y_S^i)\}_{i=1}^{N_S}$ and unlabeled target data $\mathcal{D}_\mathcal{T} = \{X_\mathcal{T}^j\}_{j=1}^{N_\mathcal{T}}$, with $Y \in \mathbb{R}^d$. Figure 1 summarizes the architecture: the encoders remain frozen, while modality projections and regression heads are trained.

4.1 TEXT CONSTRUCTION AND REFINEMENT

For image X, let $V = \phi(X)$ and $e(y) = \psi(\mathcal{P}(y))$, where $\mathcal{P}$ inserts numerical attributes into dataset-specific templates and ψ is the frozen text encoder. Source text is $T_S^i = e(Y_S^i)$. Appendix F gives the template catalog.

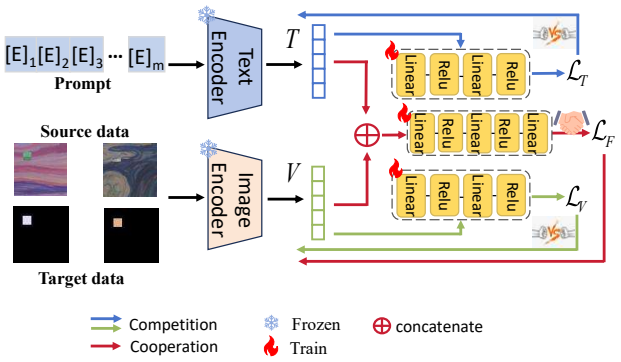


Figure 1: TCC architecture with frozen encoders, trainable modality projections, and auxiliary and fusion regression heads.

Prototype initialization. We construct source text prototypes $\{T_k^{\text{proto}}\}_{k=1}^K$, each associated with a target-value prototype Y_k^{proto}. Each text prototype averages the encodings of its prompt over multiple template forms.

The initial confidence quantile ρ controls coverage, approximately $1-\rho$ in the absence of tied scores. Similarity is a selection score, not a calibrated probability of regression accuracy.

For target feature $V_\mathcal{T}^j$, compute

$$S_{j,k} = \left\langle \frac{V_\mathcal{T}^j}{\|V_\mathcal{T}^j\|_2}, \frac{T_k^{\text{proto}}}{\|T_k^{\text{proto}}\|_2} \right\rangle, \qquad k_j^* = \arg\max_k S_{j,k}. \tag{11}$$

With a fixed tie-breaking rule, the initial pseudo-label and its confidence score are

$$\widehat{Y}_\mathcal{T}^{j,(0)} = Y_{k_j^*}^{\text{proto}}, \qquad s_j^{(0)} = \max_k S_{j,k}. \tag{12}$$

We retain samples whose scores exceed a quantile threshold,

$$\delta^{(0)} = \text{Quantile}\left(\{s_j^{(0)}\}_{j=1}^{N_\mathcal{T}}, \rho\right), \qquad 0 < \rho < 1. \tag{13}$$

Text features are defined for all target samples, while the initial selected training set is

$$T_\mathcal{T}^{j,(0)} = e(\widehat{Y}_\mathcal{T}^{j,(0)}), \tag{14}$$

$$\widehat{\mathcal{D}}_\mathcal{T}^{(0)} = \{(V_\mathcal{T}^j, T_\mathcal{T}^{j,(0)}, \widehat{Y}_\mathcal{T}^{j,(0)}) : s_j^{(0)} > \delta^{(0)}\}. \tag{15}$$

Refinement. After cycle c, the fusion predictor uses the preceding cycle's text to update pseudo-labels, then rebuilds text:

$$\widehat{Y}_{\mathcal{T}}^{j,(c)} = f_F^{(c)}([V_{\mathcal{T}}^j; T_{\mathcal{T}}^{j,(c-1)}]), \tag{16}$$

$$T_{\mathcal{T}}^{j,(c)} = e(\widehat{Y}_{\mathcal{T}}^{j,(c)}), \qquad c = 1, \ldots, C. \tag{17}$$

Subsequent cycles train on source samples and all updated target triples. Thus q_0 is prototype matching and $q_c(V) = f_F^{(c)}([V; e(q_{c-1}(V))])$. Refinement depends on both pseudo-label quality and fusion sensitivity (Theorem 2; Appendix A.4).

4.2 Branch supervision and training objective

The visual and text projections feed their auxiliary heads and the fusion head. Denoting complete branch mappings by f_V, f_T, f_F gives

$$\widehat{Y}_V = f_V(V), \qquad \widehat{Y}_T = f_T(T), \qquad \widehat{Y}_F = f_F([V; T]). \tag{18}$$

Competition refers to direct modality-specific supervision; cooperation refers to fusion of both representations. All objectives are jointly minimized. Proposition 3 in Appendix A.5 identifies implicit prediction agreement in their squared-error component, without requiring representation independence.

For a mini-batch $\mathcal{B}$ of size n, let $\bar{Y}$ denote the source label or the preceding target pseudo-label, and let $\ell_2(u, y) = \|u - y\|_2^2/d$. We use RI-Loss (Wang et al., 2026):

$$\widehat{\mathcal{L}}_{\mathrm{RI}} = \frac{1}{n} \sum_{i \in \mathcal{B}} \ell_2(\bar{Y}_i, \widehat{Y}_i) + \alpha \exp\Big[-\tau \widehat{\mathrm{HSIC}}(\overline{\mathbf{Y}}_{\mathcal{B}} - \widehat{\mathbf{Y}}_{\mathcal{B}}, \epsilon)\Big]. \tag{19}$$

Here ϵ is the random reference sequence, $\widehat{\mathrm{HSIC}}$ is empirical dependence (Gretton et al., 2007), and $\alpha, \tau > 0$. The negative exponential favors greater empirical residual–reference dependence. It does not enforce independence or by itself identify data-generating noise (Appendix A.7).

Writing $\mathcal{L}_j^{\mathrm{RI}} = \mathbb{E}_{\mathcal{B}}[\widehat{\mathcal{L}}_{\mathrm{RI}}(\overline{\mathbf{Y}}_{\mathcal{B}}, \widehat{\mathbf{Y}}_{j,\mathcal{B}})]$ for $j \in \{V, T, F\}$, the total objective is

$$\mathcal{L}_{\mathrm{total}} = \lambda_1 \mathcal{L}_V^{\mathrm{RI}} + \lambda_2 \mathcal{L}_T^{\mathrm{RI}} + \lambda_3 \mathcal{L}_F^{\mathrm{RI}}. \tag{20}$$

The expectation averages mini-batches because HSIC couples samples. Auxiliary and fusion losses both update the shared modality projections.

Inference. The output is the fusion prediction already computed in the last update,

$$\widehat{Y}_{\mathcal{T}}^{j,\mathrm{out}} = f_F^{(C)}([V_{\mathcal{T}}^j; T_{\mathcal{T}}^{j,(C-1)}]) = q_C(V_{\mathcal{T}}^j). \tag{21}$$

The preceding text is cached during refinement; inference introduces no same-step dependence on an unknown label. Algorithm 1 provides the full sequence.

5 Experiments

5.1 Experimental settings

We evaluate dSprites (Matthey et al., 2017) (scale and two positions across Color, Noise, and Scream), MPI3D (Gondal et al., 2019) (two rotations across Toy, Realistic, and Real), and BIWI (Fanelli et al., 2013) (yaw, pitch, and roll across Male and Female groups). They provide six, six, and two transfer directions. Appendix C describes the datasets and Appendix B.3 gives the network configuration; Appendix F lists prompt forms.

The modality projections map 512-dimensional encoder features to 128-dimensional embeddings; fusion uses their 256-dimensional concatenation. Training uses AdamW with a learning rate of 10^{-3} and a batch size of 64. Initialization retains the top 40% of target samples based on similarity; subsequent cycles use the updated set. Further optimization details are in Appendix B.3. We report MAE averaged over three runs. The training objective contains MSE; no monotonic relation between changes in MSE and MAE is assumed.

Table 1: MAE Comparisons on the dSprites regression tasks.

Task	Resnet-18	TCA	MCD	JDOT	AFN	DAN	DANN	RSD	GRAM	COD	UGA	TCC(our)
C → N	0.94	0.94	0.81	0.86	1.00	0.70	0.47	0.31	0.30	0.12	**0.04**	0.06
C → S	0.90	0.87	0.81	0.79	0.96	0.77	0.46	0.31	0.20	0.16	0.10	**0.07**
N → C	0.16	0.19	0.17	0.19	0.16	0.12	0.16	0.12	0.11	0.10	**0.02**	0.06
N → S	0.65	0.66	0.65	0.64	0.62	0.50	0.65	0.53	0.25	0.23	0.16	**0.05**
S → C	0.08	0.10	0.07	0.10	0.08	0.06	0.05	0.07	0.05	0.08	**0.02**	0.03
S → N	0.26	0.23	0.19	0.23	0.32	0.11	0.10	0.08	0.07	0.12	**0.02**	0.04
Avg	0.498	0.498	0.450	0.468	0.523	0.377	0.315	0.237	0.164	0.134	0.06	**0.052**

Table 2: MAE Comparisons on the MPI3D regression tasks.

Task	Resnet-18	TCA	MCD	JDOT	AFN	DAN	DANN	RSD	GRAM	COD	UGA	TCC(our)
RL → RC	0.17	0.17	0.13	0.16	0.18	0.12	0.09	0.09	0.09	0.10	0.07	**0.004**
RL → T	0.44	0.42	0.40	0.41	0.45	0.35	0.24	0.19	0.15	0.11	0.13	**0.006**
RC → RL	0.19	0.19	0.15	0.16	0.20	0.12	0.11	0.08	0.10	0.13	0.07	**0.003**
RC → T	0.45	0.42	0.45	0.41	0.46	0.27	0.41	0.15	0.14	0.15	0.11	**0.007**
T → RL	0.51	0.50	0.52	0.47	0.53	0.40	0.48	0.36	0.24	0.37	0.34	**0.004**
T → RC	0.50	0.50	0.50	0.47	0.53	0.41	0.37	0.36	0.24	0.29	0.24	**0.01**
Avg	0.377	0.373	0.358	0.353	0.390	0.278	0.283	0.205	0.160	0.192	0.16	**0.006**

Table 3: MAE Comparisons on the Biwi Kinect Head Pose regression tasks.

Task	Resnet-18	TCA	MCD	JDOT	AFN	DAN	DANN	RSD	GRAM	COD	UGA	TCC(our)
M → F	0.29	0.31	0.31	0.29	0.32	0.28	0.30	0.26	0.23	0.20	0.14	**0.05**
F → M	0.38	0.39	0.37	0.39	0.41	0.37	0.37	0.30	0.29	0.21	0.11	**0.06**
Avg	0.335	0.350	0.340	0.340	0.365	0.325	0.335	0.280	0.260	0.205	0.125	**0.055**

5.2 Comparison with adaptation baselines

We compare TCA (Pan et al., 2010), MCD (Saito et al., 2018), JDOT (Courty et al., 2017), AFN (Xu et al., 2019), DAN (Long et al., 2015), DANN (Ganin et al., 2016), regression-oriented RSD (Chen et al., 2021), DARE-GRAM (GRAM in the tables) (Nejjar et al., 2023), COD (Yang et al., 2024), and UGA (Nejjar et al., 2025).

dSprites. Table 1 reports average MAE 0.052 for TCC and 0.06 for UGA. The two methods differ across transfers: TCC gives 0.07 versus 0.10 for C→S and 0.05 versus 0.16 for N→S, whereas UGA is stronger in the other four directions. Thus, a better aggregate score does not imply that text re-encoding benefits every domain pair. The distinction matters because Theorem 2 allows both compatibility and text-construction error to vary with the transfer.

MPI3D and BIWI. TCC has the lowest listed error in all six MPI3D directions (Table 2), with an average of 0.006 compared with 0.377 for the ResNet-18 baseline. On BIWI (Table 3), it gives 0.05 and 0.06 on M→F and F→M, respectively. These benchmark comparisons evaluate complete systems. They do not separately identify the contributions of pretrained features, prototype initialization, or an individual term in the risk bound. The branch ablations below examine one part of this decomposition experimentally.

5.3 Branch ablations

Table 4 summarizes branch configurations on dSprites and MPI3D. Adding auxiliary objectives to fusion improves the reported averages in both datasets. Auxiliary-only training performs substantially worse, motivating analysis of its contribution through the shared projections in the complete system.

The detailed results in Table 5 and Appendix D cover every transfer and the BIWI study. They show task-dependent behavior; auxiliary-only errors are not a measure of representation diversity. Proposition 3 explains implicit prediction agreement in the squared-loss component. Because each retained branch uses RI-Loss, these experiments do not isolate the residual regularizer from plain MSE supervision.

Table 5 gives the transfer-wise MPI3D ablation. Adding auxiliary objectives to fusion reduces RL→T from 0.07 to 0.006 and T→RL from 0.01 to 0.004. Other directions show smaller changes or favor fusion alone. For example, RL→RC changes from 0.003 to 0.004. The average improvement, therefore, reflects a balance across directions rather than a uniformly beneficial auxiliary constraint. Appendix D provides the dSprites and BIWI transfer-wise tables and further interpretation.

Table 4: Reported average MAE for branch configurations. See Table 5 and Appendix D for individual transfers.

Objective	dSprites	MPI3D
Text only	0.748	0.380
Visual only	1.217	0.485
Fusion only	0.093	0.016
Auxiliary only	1.382	0.563
All branches	**0.052**	**0.006**

Table 5: MPI3D branch ablations (MAE).

$\mathcal{L}_T^{\mathrm{RI}}$	$\mathcal{L}_V^{\mathrm{RI}}$	$\mathcal{L}_F^{\mathrm{RI}}$	**RL → RC**	**RL → T**	**RC → RL**	**RC → T**	**T → RL**	**T → RC**	**Avg**
√			0.34	0.08	0.25	0.79	0.35	0.47	0.38
	√		0.44	0.53	0.37	0.48	0.53	0.56	0.485
		Cooperation	0.003	0.07	0.004	0.004	0.01	0.004	0.016
Competition			0.45	0.47	0.79	0.64	0.57	0.46	0.563
Competition		**Cooperation**	0.004	0.006	0.003	0.007	0.004	0.01	**0.006**

5.4 Representation and Optimization Diagnostics

We use proxy $\mathcal{A}$-distance, $\mathcal{D}_{\mathcal{A}} = 2(1 - 2\epsilon(h^*))$, where $\epsilon(h^*)$ is the error of a domain classifier with equally weighted source and target samples. Figure 2 shows lower values for TCC on C→N and C→S. This diagnostic concerns marginal representations; it neither estimates the reference-text discrepancy in Theorem 2 nor certifies label compatibility.

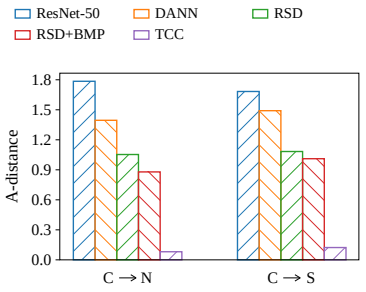


Figure 2: Proxy $\mathcal{A}$-distance on two dSprites transfers. Lower values indicate less distinguishable feature marginals.

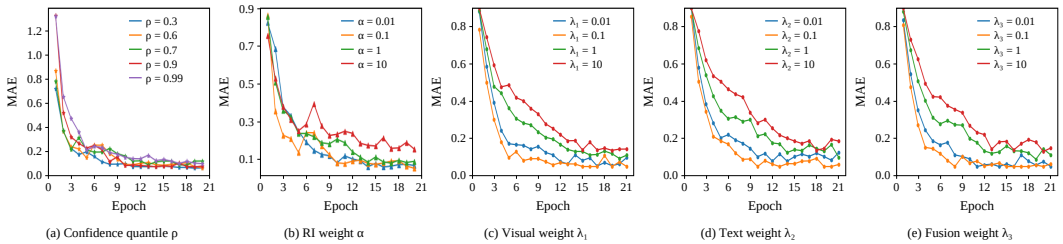
Appendix E provides training trajectories and further interpretation of the parameter sweeps in Figure 3. The displayed objective curves stabilize. These observations describe the tested settings; they do not imply contraction of pseudo-label error or a population guarantee for RI-Loss.

5.5 Parameter Sensitivity

Figure 3 groups the sensitivity analyses on dSprites C→N: confidence selection, residual-informed weighting, and relative branch supervision. All panels show evaluation MAE against training epochs.

Confidence selection. A larger initial quantile ρ retains fewer target samples, while later cycles use the full updated set. Figure 3(a) shows similar final errors across the tested settings, with different transient behavior. This coverage–selection tradeoff does not calibrate similarity as numerical pseudo-label accuracy. Appendix A.4 separates the selected and remaining samples' contributions to full target risk.

Residual and branch weights. The coefficient α scales the residual-informed term in Eq. 19; $\lambda_1, \lambda_2, \lambda_3$ weight visual, text, and fusion supervision. Figures 3(b)–(e) show that these choices affect optimization. Smaller tested branch weights generally yield lower final MAE than 10 in this transfer. The sweeps neither identify universal weights nor isolate RI-Loss from plain MSE. Since



(a) Confidence quantile ρ (b) RI weight α (c) Visual weight λ_1 (d) Text weight λ_2 (e) Fusion weight λ_3

Figure 3: Parameter sensitivity on dSprites C→N. Panels (a)–(e) compare the confidence quantile, RI weight, and visual, text, and fusion branch weights, respectively.

the negative exponential favors empirical residual–reference dependence, they also do not establish residual independence or noise recovery (Appendix A.7).

5.6 Discussion: connecting the analysis and experiments

The theoretical results distinguish representation utility from information gain. Because target text is constructed from visual predictions, its benefit must be assessed relative to the trained hypothesis class rather than attributed to additional information conditional on the complete visual input. The branch ablations empirically examine this choice of representation by comparing visual, textual, and fusion configurations to assess whether the available encodings are useful to the learned regressors. Such comparisons do not, by themselves, identify whether a gain comes from pretrained representations, auxiliary supervision, or the refinement procedure.

The parameter sweeps also have distinct theoretical interpretations. Confidence selection changes the subset used for initialization, so its effect combines retained coverage with the quality of the selected pseudo-labels. Branch weights change both prediction fitting and the implicit agreement term in the squared-loss identity (Appendix A.5). Agreement can nevertheless occur around an incorrect prediction. Likewise, stable training objectives in Appendix E describe optimization behavior, whereas the refinement analysis concerns numerical prediction error. These quantities should be evaluated separately when assessing whether successive text updates improve transfer.

The BIWI ablations in Appendix D illustrate this task dependence. Visual-only supervision outperforms text-only supervision on both head-pose transfers, reversing their ordering on the synthetic benchmarks. Yet the complete objective achieves an MAE of 0.05 on M→F and 0.06 on F→M, both below the listed alternatives in each direction. Thus, the performance of an isolated textual branch does not determine the utility of text within the complete framework. This supports the evaluation of modality-specific supervision together with fusion.

Together, these diagnostics address feature distinguishability, initialization, and branch supervision, but do not measure all terms in the target-risk bound. Lower marginal discrepancy need not imply a compatible regression rule, and similar final errors across confidence settings do not certify similarity-based calibration.

6 Conclusion

We study the reliability of generated text for unsupervised domain adaptation regression. Our analysis distinguishes deterministic re-encoding from additional information, separates the factors governing target risk, and characterizes branch agreement and conditional refinement. These insights motivate retaining visual evidence, accounting for pseudo-label reliability, and jointly supervising modality-specific and fusion predictions. Guided by this analysis, TCC combines three regression branches with a residual-informed objective and updates the target text by fusing predictions using the preceding text. Experiments on dSprites, MPI3D, and BIWI assess TCC through baseline comparisons, ablations, and diagnostics. Future work will measure numerical text sensitivity and isolate the effects of pretraining, initialization, and residual regularization to clarify when re-encoding improves transfer.

AI USE STATEMENT

Generative AI tools assisted the revision of this manuscript, including language editing, organization, refinement of theoretical arguments and proofs, notation and consistency checks, and figure formatting. They were not used to generate the reported experimental results. The authors retain full responsibility for the accuracy and validity of the theoretical claims, experimental results, and final manuscript, including all AI-assisted content.

REPRODUCIBILITY STATEMENT

Section 4 describes the TCC framework and optimization objectives, and Section 5 reports the experimental setup and evaluation metric. Appendix A provides the assumptions, additional results, and proofs supporting the theoretical analysis. Appendix B specifies the training and prediction procedure, including pseudo-label initialization, iterative text updates, and the network configuration. Dataset descriptions, detailed branch ablations, optimization diagnostics, and prompt templates are provided in Appendices C–F. These materials document the distinction between the training objective and evaluation MAE, as well as the use of the preceding round's text in fusion-based prediction. The source code will be made publicly available to support reproduction of the reported results.

REFERENCES

Shuanghao Bai, Min Zhang, Wanqi Zhou, Siteng Huang, Zhirong Luan, Donglin Wang, and Badong Chen. Prompt-based distribution alignment for unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 729–737, 2024.

Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In *ICML*, pp. 1749–1759, 2021.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017.

Thomas M Cover, Joy A Thomas, and John Kieffer. *Elements of information theory*, volume 2. wiley New York, 1991.

Yifei Ding, Peng Ding, Xiaoli Zhao, Yudong Cao, and Minping Jia. Transfer learning for remaining useful life prediction across operating conditions based on multisource domain adaptation. *IEEE/ASME transactions on mechatronics*, 27(5):4143–4152, 2022.

Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International journal of computer vision*, 101(3):437–458, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611, 2016. URL https://api.semanticscholar.org/CorpusID:206596513.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32, 2019.

Daniel Greenfeld and Uri Shalit. Robust learning with the hilbert-schmidt independence criterion. In *International Conference on Machine Learning*, pp. 3759–3768. PMLR, 2020.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Yunfan Li, Peng Hu, Dezhong Peng, Jiancheng Lv, Jianping Fan, and Xi Peng. Image clustering with external guidance. *arXiv preprint arXiv:2310.11989*, 2023.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017.

Ismail Nejjar, Qin Wang, and Olga Fink. Dare-gram: Unsupervised domain adaptation regression by aligning inverse gram matrices. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11744–11754. IEEE, 2023.

Ismail Nejjar, Gaetan Frusque, Florent Forest, and Olga Fink. Uncertainty-guided alignment for unsupervised domain adaptation in regression. *Reliability Engineering & System Safety*, pp. 112143, 2025.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2155–215509. IEEE, 2018.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3723–3732. IEEE, 2018.

Katherine Tsai, Stephen R Pfohl, Olawale Salaudeen, Nicole Chiou, Matt Kusner, Alexander D'Amour, Sanmi Koyejo, and Arthur Gretton. Proxy methods for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3961–3969. PMLR, 2024.

Jieting Wang, Xiaolei Shang, Feijiang Li, and Furong Peng. Ri-loss: A learnable residual-informed loss for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26277–26284, 2026.

Yaoming Wang, Yangzhou Jiang, Jin Li, Bingbing Ni, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Teng Li. Contrastive regression for domain adaptation on gaze estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19354–19363. IEEE, 2022.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *2019 IEEE/CVF international conference on computer vision (ICCV)*, pp. 1426–1435. IEEE, 2019.

Hao-Ran Yang, Chuan-Xian Ren, and You-Wei Luo. Cod: Learning conditional invariant representation for domain adaptation regression. In *European conference on computer vision*, pp. 108–125. Springer, 2024.

APPENDIX

The supplementary material is organized as follows. Each section provides the definitions, procedure, or experimental context needed to interpret the associated results.

- Appendix A: Additional theoretical results and proofs (p. 12).
- Appendix B: Training and prediction procedure (p. 16).
- Appendix C: Datasets and domain examples (p. 17).
- Appendix D: Detailed branch ablations (p. 18).
- Appendix E: Optimization and parameter analyses (p. 19).
- Appendix F: Prompt template catalogue (p. 20).

A ADDITIONAL THEORETICAL RESULTS AND PROOFS

This appendix expands Section 3. We first establish the information identity and illustrate restricted-model benefits, then prove the target-risk bound and analyze refinement. The final parts explain branch agreement, the entropy lower bound, and the scope of RI-Loss. Throughout, risks use the coordinate-normalized losses defined below.

Symbol	Meaning
$P_S, P_{\mathcal{T}}$	Source and target joint distributions of images and regression labels.
$Y \in \mathbb{R}^d$	Regression target vector; d is the number of evaluated coordinates.
$V = \phi(X), U = a(V)$	Frozen visual feature and a trainable compressed visual representation.
$e = \psi \circ \mathcal{P}$	Prompt construction followed by the pretrained text encoder.
q_c, $f_E^{(c)}$	Pseudo-label mapping and trained fusion predictor at cycle c.
T^*, T	Reference text $e(Y)$ and deployed pseudo-label text $e(q(V))$.
$\mathcal{R}_2^*(W)$	Bayes MSE given input W, minimized over measurable predictors.
$R_D^*(f), \widetilde{R}_{\mathcal{T}}(f, q)$	Reference-text MAE in domain D and deployed target MAE.
$Q_D^*, \mathcal{H}$	Distribution of $(V, e(Y))$ and the class of regression predictors.
$\Lambda_{\mathcal{H}}$	Best joint source/target reference-text risk within $\mathcal{H}$.
$\Delta_{\mathcal{T}}(q), L_f$	Expected text-construction error and fusion sensitivity to text.
$\bar{Y}, p_j, a_j$	Available training label, branch prediction, and normalized branch weight.

Table 6: Notation used in the analysis. A superscript $*$ on R_D^* denotes reference text; only $\mathcal{R}_2^*$ denotes a Bayes optimum.

A.1 PROOF OF PROPOSITION 1

Proof. Write $m(V) = \mathbb{E}[Y \mid V]$ and $m'(V,T) = \mathbb{E}[Y \mid V,T]$. The conditional expectation minimizes expected squared Euclidean error, so

$$\mathcal{R}_2^*(V) = \frac{1}{d}\mathbb{E}\|Y - m(V)\|_2^2, \qquad \mathcal{R}_2^*(V,T) = \frac{1}{d}\mathbb{E}\|Y - m'(V,T)\|_2^2.$$

Using $Y - m(V) = (Y - m'(V,T)) + (m'(V,T) - m(V))$, expand the first squared norm. Its cross term is zero because

$$\mathbb{E}\langle Y - m'(V,T), m'(V,T) - m(V)\rangle = \mathbb{E}\langle \mathbb{E}[Y - m'(V,T) \mid V,T], m'(V,T) - m(V)\rangle = 0.$$

Subtracting the two risks proves Eq. 1. If $T = e(q(V))$ is deterministic and measurable, then $\sigma(V,T) = \sigma(V)$, and consequently $m' = m$ almost surely. Conditional on $V = v$, the variable T is a point mass at $e(q(v))$. The conditional joint law of (T,Y) therefore factors into its conditional marginals, giving $I(T;Y \mid V) = 0$. This argument also applies conditional on fixed encoders, prototypes, and learned parameters. It does not treat random training outputs as an extra test-time observation. □

A.2 A REPRESENTATION BENEFIT WITHOUT ADDITIONAL INFORMATION

Consider $V \sim \mathrm{Unif}[-1, 1]$, $Y = V^2$, and the deterministic re-encoding $T = V^2$. For the restricted visual class $\{aV + b : a, b \in \mathbb{R}\}$, symmetry gives the best predictor $b = 1/3$, $a = 0$, with MSE

$$\mathbb{E}(V^2 - 1/3)^2 = \mathbb{E}[V^4] - (\mathbb{E}[V^2])^2 = \frac{1}{5} - \frac{1}{9} = \frac{4}{45}.$$

The linear class on (V, T) contains $f(V, T) = T$ and achieves zero error, although $I(T; Y \mid V) = 0$ and both unrestricted Bayes risks are zero. This example illustrates an approximation benefit; it is not an assumption about the numerical behavior of a pretrained text encoder.

For a fixed compressed representation $U = a(V)$, the same projection argument gives

$$\mathcal{R}_2^*(U) - \mathcal{R}_2^*(U, T) = \frac{1}{d}\mathbb{E}\|\mathbb{E}[Y \mid U, T] - \mathbb{E}[Y \mid U]\|_2^2.$$

The right-hand side can be positive when T retains information discarded by a. This possibility concerns the compressed input U, whereas Eq. 2 concerns the full input V. Neither statement guarantees that the learned projections or the given prompt bank realize a strict improvement.

A.3 PROOF OF THEOREM 2

Proof. First compare target prediction with pseudo-label-derived text and reference text. The triangle inequality for ℓ_1 and Eq. 6 imply

$$\begin{aligned}\widetilde{R}_{\mathcal{T}}(f, q) &\le R_{\mathcal{T}}^*(f) + \mathbb{E}_{\mathcal{T}}\ell_1(f(V, e(q(V))), f(V, e(Y))) \\ &\le R_{\mathcal{T}}^*(f) + L_f\Delta_{\mathcal{T}}(q). \end{aligned} \tag{22}$$

For any $g \in \mathcal{H}$, apply the triangle inequality again, followed by the definition of discrepancy:

$$\begin{aligned}R_{\mathcal{T}}^*(f) &\le \mathbb{E}_{Q_{\mathcal{T}}^*}\ell_1(f(Z^*), g(Z^*)) + R_{\mathcal{T}}^*(g) \\ &\le \mathbb{E}_{Q_S^*}\ell_1(f(Z^*), g(Z^*)) + \mathrm{disc}_{\mathcal{H}}(Q_S^*, Q_{\mathcal{T}}^*) + R_{\mathcal{T}}^*(g) \\ &\le R_S^*(f) + R_S^*(g) + \mathrm{disc}_{\mathcal{H}}(Q_S^*, Q_{\mathcal{T}}^*) + R_{\mathcal{T}}^*(g).\end{aligned}$$

Taking the infimum over g and using Eq. 22 proves the result; attainment of the infimum is unnecessary. This is a specialization of the discrepancy-based adaptation argument of Mansour et al. (2009), together with a comparison of the two text-construction rules. The proof uses a metric loss, not a triangle inequality for squared error. □

Interpretation of the reference distribution. The law Q_D^* includes label-derived text. It differs from the law of the deployed target input $(V, e(q(V)))$. Keeping these two laws separate is essential even if the original image-label task satisfies covariate shift. A source regressor may exploit a nearly invertible label encoding, making R_S^* small without correcting target pseudo-labels. The target-reference risk is consequently not an achievable test-time baseline. Neither it nor $\Lambda_{\mathcal{H}}$ can be evaluated from unlabeled target data alone. Theorem 2 does not equate these quantities with the visual domain-classification statistic reported in the experiments.

Population risk versus a finite training sample. The theorem holds for each fixed predictor and text generator under the stated conditions. It is not a concentration bound on the empirical training objective. A finite-sample extension must control the relevant predictor classes and the data dependence of prototypes, confidence selection, and successive pseudo-label generators. Conditioning on a fitted model alone does not make its reused training samples independent of that model. No such independence assumption or unproved sample-complexity term is used here.

A.4 SQUARED-ERROR SENSITIVITY AND CONDITIONAL REFINEMENT

For any random vector A, define $\|A\|_{L_d^2} = (\mathbb{E}\|A\|_2^2/d)^{1/2}$. Decompose the deployed residual as

$$f(V, e(q(V))) - Y = \big(f(V, e(Y)) - Y\big) + \big(f(V, e(q(V))) - f(V, e(Y))\big).$$

The triangle inequality in L_d^2 and the assumed text sensitivity yield

$$\sqrt{\widetilde{R}_{\mathcal{T},2}(f,q)} \le \sqrt{R^*_{\mathcal{T},2}(f)} + \left(\frac{1}{d}\mathbb{E}_{\mathcal{T}}\|f(V,e(q(V))) - f(V,e(Y))\|_2^2\right)^{1/2}$$
$$\le \sqrt{R^*_{\mathcal{T},2}(f)} + L_{f,2}\left(\mathbb{E}_{\mathcal{T}}\|e(q(V)) - e(Y)\|_2^2\right)^{1/2},$$

which proves Eq. 8. No smoothness of e is needed for this statement.

We now prove Corollary 4, with the error and sensitivity quantities defined in Section 3.4.

Proof. Substitute the assumed bound on e into Eq. 8 with $f = f_F^{(c)}$ and $q = q_{c-1}$. Apply the resulting recurrence successively and sum the geometric series. □

The corollary describes a sufficient condition, not a property established for the learned models. Prompt tokenization, rounding, and nearest-prototype changes can invalidate a useful global constant L_e. Without this condition, the direct embedding-error term in Eq. 8 remains applicable. Even with $\kappa < 1$, the recurrence bounds an error level and does not imply that every realized cycle improves accuracy. Wrong pseudo-labels may be reproduced by a textual decoder or reinforced by all branches.

When does a refinement step actually improve MSE? An exact comparison is possible without smoothness assumptions. Set $D_c(V) = q_c(V) - q_{c-1}(V)$. Expanding the squared residual gives

$$\varepsilon_c^2 - \varepsilon_{c-1}^2 = \frac{1}{d}\mathbb{E}_{\mathcal{T}}\|D_c(V)\|_2^2 + \frac{2}{d}\mathbb{E}_{\mathcal{T}}\langle q_{c-1}(V) - Y, D_c(V)\rangle. \tag{23}$$

Strict improvement therefore requires the update to oppose the previous prediction error strongly enough to offset its squared magnitude. Merely decoding the preceding pseudo-label gives $D_c = 0$ and no improvement. This identity supplies a diagnostic for evaluating refinement on held-out labels; it is not an additional training objective and does not make target labels available to the algorithm.

Confidence selection. Let A_ρ be the event that an initialized target sample passes the similarity threshold, and let $\pi_\rho = P_{\mathcal{T}}(A_\rho)$. For $0 < \pi_\rho < 1$ and any deployed nonnegative loss, the full target risk is exactly

$$\widetilde{R}_{\mathcal{T}} = \pi_\rho\,\mathbb{E}_{\mathcal{T}}[\ell \mid A_\rho] + (1-\pi_\rho)\,\mathbb{E}_{\mathcal{T}}[\ell \mid A_\rho^c].$$

Thus, fitting the selected subset does not by itself control the unselected subset. A quantile of cosine similarity is a selection rule, not a calibrated probability of numerical accuracy. The training procedure in Appendix B applies this rule at initialization and uses all updated target samples in subsequent cycles. Our risk statements concern the full target distribution.

A.5 Branch supervision and prediction agreement

We prove the decomposition stated in Proposition 3, using the same predictions and normalized weights.

Proof. For each branch, write $p_j - \bar{Y} = (p_j - \bar{p}) + (\bar{p} - \bar{Y})$. Expanding the squared norms and summing with weights a_j gives

$$\sum_j a_j\|p_j - \bar{Y}\|_2^2 = \sum_j a_j\|p_j - \bar{p}\|_2^2 + \|\bar{p} - \bar{Y}\|_2^2$$
$$+ 2\left\langle \sum_j a_j(p_j - \bar{p}), \bar{p} - \bar{Y}\right\rangle.$$

The last term vanishes by the definition of $\bar{p}$. Dividing by d proves Eq. 9. Equivalently,

$$\sum_j a_j\|p_j - \bar{p}\|_2^2 = \sum_{i<j} a_i a_j\|p_i - p_j\|_2^2.$$

The identity holds sample-wise and therefore also after averaging over source and pseudo-labeled target samples. Multiplying by $\lambda_1 + \lambda_2 + \lambda_3$ recovers the unnormalized squared-error component of the training objective. □

This decomposition is not a guarantee of statistical independence or diverse representations. Direct auxiliary supervision can influence fusion through the shared modality projections, but the identity alone does not establish a reduction of target error or a benefit of every positive branch weight. The nonlinear residual-reference term in RI-Loss remains outside this algebraic decomposition.

A.6 Conditional-entropy lower bound and its limitations

The entropy bound is a standard consequence of the Gaussian maximum-entropy property and Jensen's inequality (Cover et al., 1991). We state the regularity conditions explicitly and use natural logarithms.

Lemma 5 (Conditional-entropy bound for scalar regression)**.** *Let $Y \in \mathbb{R}$ satisfy $\mathbb{E}[Y^2] < \infty$. Assume that $Y \mid (V, T)$ admits a density almost surely, its conditional differential entropy is finite, and the conditional-entropy expectations below are well defined. Then*

$$\mathcal{R}_2^*(V,T) \geq \frac{1}{2\pi \mathrm{e}} \exp\big(2h(Y \mid V,T)\big). \tag{24}$$

If $h(Y \mid V)$ is also finite, then $h(Y \mid V,T) = h(Y \mid V) - I(T;Y \mid V)$. If T is discrete and the relevant conditional Shannon entropies are finite, then $I(T;Y \mid V) = H(T \mid V) - H(T \mid V,Y)$. Equality in Eq. 24 requires and is attained by conditionally Gaussian laws with the same positive conditional variance almost surely.

Proof. The Bayes estimator is $g^*(V,T) = \mathbb{E}[Y \mid V,T]$, so $\mathcal{R}_2^*(V,T) = \mathbb{E}[\mathrm{Var}(Y \mid V,T)]$. For almost every (v,t), the Gaussian maximum-entropy property gives

$$h(Y \mid V=v, T=t) \leq \frac{1}{2}\log\big(2\pi\mathrm{e}\,\mathrm{Var}(Y \mid V=v, T=t)\big).$$

Taking expectations and using the concavity of the logarithm yields

$$\begin{aligned} h(Y \mid V,T) &\leq \frac{1}{2}\mathbb{E}\log\big(2\pi\mathrm{e}\,\mathrm{Var}(Y \mid V,T)\big) \\ &\leq \frac{1}{2}\log\big(2\pi\mathrm{e}\mathbb{E}\,\mathrm{Var}(Y \mid V,T)\big) = \frac{1}{2}\log\big(2\pi\mathrm{e}\mathcal{R}_2^*(V,T)\big). \end{aligned}$$

Rearrange to obtain Eq. 24. The entropy identities follow from the definition of conditional mutual information under the stated finiteness conditions. Equality requires equality in both the Gaussian maximum-entropy inequality and Jensen's inequality, giving the stated Gaussian and constant-variance conditions. □

Information about variance need not improve a conditional mean. Let V be constant, let T be equiprobable on $\{0,1\}$, and let $Y \mid T=0 \sim \mathcal{N}(0,1)$ and $Y \mid T=1 \sim \mathcal{N}(0,4)$. The two conditional distributions differ, so $I(T;Y) > 0$. However, their conditional means are both zero and $\mathcal{R}_2^*(V,T) = \mathcal{R}_2^*(V) = (1+4)/2 = 2.5$. Thus, positive conditional mutual information is not sufficient for a strict MSE improvement, and lowering a lower bound does not prove lower prediction error. Proposition 1 specifies the exact condition for squared loss.

Discrete targets and encoded text. A finite grid of regression labels is not a continuous density to which Lemma 5 can automatically be applied. The main risk bounds and projection identities do apply to discrete, continuous, or mixed labels with the required moments. For multiple outputs, the scalar entropy bound can be applied coordinate-wise only where its regularity assumptions hold. The random variable T in the main method is an encoded vector; a discrete-entropy expansion is valid only if the particular encoded variable has discrete support with the stated finite entropies. A token sequence and its encoded vector should not be interchanged without checking this distinction.

A.7 Statistical scope of the RI-Loss term

RI-Loss is the *Residual-Informed Loss* of Wang et al. (2026). For $r = Y - \widehat{Y}$ and $s = \widehat{\mathrm{HSIC}}(r,\epsilon)$, its additional term has derivative

$$\frac{\partial}{\partial s}\big(\alpha\exp(-\tau s)\big) = -\alpha\tau\exp(-\tau s) < 0 \quad (\alpha,\tau > 0).$$

It therefore favors higher empirical dependence between the paired residual and reference samples. HSIC is a dependence measure; with suitable characteristic kernels, zero population HSIC characterizes independence (Gretton et al., 2007). Dependence of two variables is distinct from similarity of their marginal distributions.

In particular, fix a predictor f and suppose a fresh reference ϵ is independent of (V, T, Y) and is not an input to f. Since $r = Y - f(V, T)$ is measurable with respect to (V, T, Y), r and ϵ are independent and their population HSIC is zero. Empirical dependence induced during training does not by itself establish that residuals equal the data-generating noise, or that the predictor is invariant across domains. Results on residual–input independence, such as those of Greenfeld & Shalit (2020), concern a different dependence objective. Likewise, a concentration result for an HSIC estimator is not automatically a guarantee of a decrease in target regression risk. The interpretation of the RI term depends on these distinctions.

B TRAINING AND PREDICTION PROCEDURE

Algorithm 1 makes the text input to the fusion teacher explicit. Initial target training uses the confidence-selected subset; subsequent cycles use all updated target triples. The encoders remain frozen throughout, and each cycle minimizes the branch objective in Eq. 20. The final predictions are those computed by the last fusion teacher with the preceding round's text, as in Eq. 21.

B.1 PROTOTYPE CONSTRUCTION AND INITIAL SELECTION

Section 4.1 defines prototype matching, confidence scores, and the initial selected set. This selection is used only for the first cycle. Each prototype pairs a numerical regression vector with a template-averaged text feature, and a fixed tie-breaking rule makes the initialization deterministic once the bank is fixed.

The score threshold controls coverage rather than certifying accuracy. Every target sample receives a text feature, including samples outside the initial training subset. This distinction is necessary for the first fusion update, which is applied to the entire target set. After each cycle, the updated numerical predictions are re-encoded and all target triples become available for the next cycle.

B.2 CYCLE-WISE TRAINING AND PREDICTION

The following procedure distinguishes the text used to produce a prediction from the text rebuilt after that prediction. All target samples receive an initial text feature, including those excluded from the first training subset. Later updates rebuild the full target training set.

B.3 NETWORK AND IMPLEMENTATION CONFIGURATION

Network. Our model employs a three-branch multimodal regression architecture with a text branch, an image branch, and a shared regression head. Both branches project 512-dimensional inputs into 128-dimensional embeddings via two fully connected layers with ReLU activations. The text and image embeddings are concatenated into a 256-dimensional joint representation and passed to a shared regression head consisting of three fully connected layers to generate the final predictions. The modality-specific projections are shared between their auxiliary prediction heads and the fusion pathway, so both auxiliary and fusion objectives update them. The distinction between these projected representations and the original encoder features is used in Section 3.1.

Optimization. Our model is trained using the AdamW optimizer. The initial learning rate is set to 1e-3, and the proportion of loss λ is set to 0.1. During training, we set the batch size to 64 and trained for 20 epochs. Initialization retains the top 40% of target samples by confidence; subsequent cycles use the updated target set in Algorithm 1. In addition, we enhance training stability by clipping the gradient norm (max_norm=1.0) to prevent gradient explosion. All experiments were implemented in PyTorch 2.5.1, and accelerated computations were performed on NVIDIA RTX A6000 48GB GPUs with CUDA 11.8.

Algorithm 1 TCC training and target prediction

Require: Source samples $\mathcal{D}_S$; unlabeled target samples $\mathcal{D}_\mathcal{T}$; frozen encoders ϕ, ψ; prompt map $\mathcal{P}$; paired prototypes $\{(T_k^{\text{proto}}, Y_k^{\text{proto}})\}_{k=1}^K$; quantile ρ; cycles $C \geq 1$; epochs E.

Ensure: Final fusion model $f_F^{(C)}$ and target predictions $\{\widehat{Y}_\mathcal{T}^{j,\text{out}}\}$.

Set $e = \psi \circ \mathcal{P}$.
Construct $\mathcal{D}_S^e = \{(\phi(X_S^i), e(Y_S^i), Y_S^i)\}_{i=1}^{N_S}$.
Extract $V_\mathcal{T}^j = \phi(X_\mathcal{T}^j)$ for every target sample.
Compute prototype scores, $\widehat{Y}_\mathcal{T}^{j,(0)}$, and $s_j^{(0)}$ using Eqs. 11–12.
Construct $T_\mathcal{T}^{j,(0)} = e(\widehat{Y}_\mathcal{T}^{j,(0)})$ for every target sample.
Form $\widehat{\mathcal{D}}_\mathcal{T}^{(0)}$ using the quantile threshold in Eqs. 13–15.
for $c = 1, \ldots, C$ **do**
    for epoch $= 1, \ldots, E$ **do**
        for each mini-batch from $\mathcal{D}_S^e$ and $\widehat{\mathcal{D}}_\mathcal{T}^{(c-1)}$ **do**
            Compute the visual, text, and fusion predictions in Eq. 18.
            Update trainable parameters using $\mathcal{L}_{\text{total}}$ in Eq. 20.
        end for
    end for
    Denote the resulting fusion predictor by $f_F^{(c)}$.
    for $j = 1, \ldots, N_\mathcal{T}$ **do**
        $\widehat{Y}_\mathcal{T}^{j,(c)} \leftarrow f_F^{(c)}([V_\mathcal{T}^j; T_\mathcal{T}^{j,(c-1)}])$.
        $T_\mathcal{T}^{j,(c)} \leftarrow e(\widehat{Y}_\mathcal{T}^{j,(c)})$.
    end for
    $\widehat{\mathcal{D}}_\mathcal{T}^{(c)} \leftarrow \{(V_\mathcal{T}^j, T_\mathcal{T}^{j,(c)}, \widehat{Y}_\mathcal{T}^{j,(c)})\}_{j=1}^{N_\mathcal{T}}$.
end for
Set $\widehat{Y}_\mathcal{T}^{j,\text{out}} = \widehat{Y}_\mathcal{T}^{j,(C)}$ for each target sample.
return $f_F^{(C)}$, the corresponding cached text inputs $\{T_\mathcal{T}^{j,(C-1)}\}$, and the final target predictions.

C Datasets and domain examples

This appendix complements the experimental settings in Section 5. It describes the domains and predicted attributes and illustrates their appearance changes. Network and training details are provided in Appendix B.3. The dataset sizes below describe the benchmark collections.

C.1 dSprites

dSprites Matthey et al. (2017) is a synthetic 2D dataset with five independent latent factors. Following standard practice, we treat its three variants—Color (**C**), Scream (**S**), and Noise (**N**)—as distinct domains, each containing 737,280 images, shown in Figure 4. The dataset is used for regression-based domain adaptation on scale and position (X, Y) attributes; the orientation factor is excluded. We evaluate six transfer directions: **C**→**N**, **C**→**S**, **N**→**C**, **N**→**S**, **S**→**C**, and **S**→**N**.



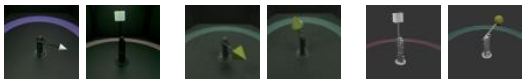
(a) Noise **(b) Color** **(c) Scream**

Figure 4: Example images from the three dSprites domains.

The scale and two position coordinates define a three-dimensional regression output. The six directions distinguish both the source and target appearance: reversing a direction changes which domain supplies labels. Figure 4 illustrates appearance changes while retaining the geometric attributes of interest. These shifts test whether the numerical prediction rule transfers across backgrounds and color patterns.

C.2 MPI3D

MPI3D Gondal et al. (2019) contains 1,036,800 images of 3D objects across three domains: Toy (**T**), Realistic (**RC**), and Real (**RL**), as shown in Figure 5. Captured in a controlled robotic setting, the dataset includes multiple factors of variation. We evaluate six transfer tasks: **RL**→**RC**, **RL**→**T**, **RC**→**T**, **RC**→**RL**, **T**→**RL**, **T**→**RC**, focusing on regression of vertical and horizontal rotations.



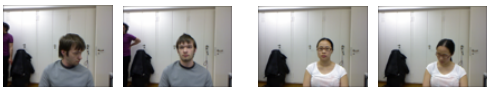
(a) Real **(b) Realistic** **(c) Toy**

Figure 5: Example images from the three MPI3D domains.

The two evaluated attributes define a two-dimensional output. The synthetic and real domains differ in rendering and image appearance, while the task remains prediction of the stated horizontal and vertical rotations. Figure 5 shows representative images; the transfer tables use RC for Realistic and RL for Real throughout.

C.3 BIWI

Biwi Kinect Head Pose (BIWI) Fanelli et al. (2013) contains over 15,000 head-pose images from 20 subjects (6 Females (**F**), 14 Males (**M**)) captured with a Kinect sensor. Regression targets are yaw, pitch, and roll angles. Cross-gender transfer is evaluated under **M**→**F** and **F**→**M**.



(a) Male **(b) Female**

Figure 6: Example images from the two BIWI domains.

The three pose angles form the output vector. The two directions exchange which subject group provides source supervision. Figure 6 illustrates the domains; it is a qualitative view of appearance, whereas Table 3 reports prediction performance. As in the other benchmarks, target labels are not inputs to text construction.

D Detailed branch ablations

This section expands the branch comparison in Section 5.3. The retained objectives are the text (T), visual (V), and fusion (F) RI losses in Eq. 20. "Competition" denotes the two auxiliary objectives together, and "Cooperation" denotes fusion supervision. The tables give transfer-wise MAE and the reported averages. These comparisons vary branch supervision; they do not independently remove the HSIC term from a retained branch.

D.1 MPI3D

Table 5 reports six transfer directions. Fusion alone has an average error of 0.016, compared with 0.006 for the complete objective. The complete model is not the best in every direction: on RL→RC, for example, fusion alone yields 0.003, while the full model yields 0.004. The aggregate comparison, therefore, supports the combined objective while retaining variation across transfers.

Table 7: dSprites branch ablations (MAE).

$\mathcal{L}_T^{\mathrm{RI}}$	$\mathcal{L}_V^{\mathrm{RI}}$	$\mathcal{L}_F^{\mathrm{RI}}$	**C → N**	**C → S**	**N → C**	**N → S**	**S → C**	**S → N**	**Avg**
√			0.57	1.71	0.25	0.48	0.31	1.17	0.748
	√		1.16	1.43	1.28	1.24	0.97	1.22	1.217
		Cooperation	0.22	0.15	0.05	0.05	0.03	0.06	0.093
Competition			1.24	1.51	1.56	1.42	1.35	1.21	1.382
Competition		**Cooperation**	0.06	0.07	0.06	0.05	0.03	0.04	**0.052**

D.2 dSprites

Table 7 compares the same configurations on appearance shifts. The complete objective reduces the reported average from 0.093 for fusion alone to 0.052. Its greatest changes are on C→N and

C→S; some directions are unchanged. High error rates in auxiliary-only configurations should not be interpreted as evidence of diverse or independent representations.

D.3 BIWI

Table 8 reports the two head-pose transfers. The complete objective gives 0.05 on M→F and 0.06 on F→M, below the listed alternatives in both directions. Visual-only supervision is stronger than text-only supervision on this benchmark. This differs from the ordering on the synthetic datasets and illustrates that auxiliary-branch performance is task-dependent.

Table 8: BIWI branch ablations (MAE). Columns T, V, F indicate retained objectives.

T	V	F	**M → F**	**F → M**	**Avg**
√			0.783	0.565	0.674
	√		0.13	0.14	0.135
		√	0.23	0.12	0.125
√	√		0.45	0.39	0.42
√	√	√	0.05	0.06	**0.055**

The shared projections allow auxiliary gradients to affect the final predictor. The agreement identity in Appendix A.5 does not by itself establish correct target predictions or isolate the effect of RI-Loss.

E Optimization and Parameter Analyses

We provide the detailed diagnostics summarized in Section 5.4. The plots describe the displayed dSprites transfers. Training-objective values and evaluation MAE are different quantities; the plots are interpreted according to their labeled axes.

E.1 Training Trajectories

Figure 7 plots the training objective over epochs on C→N, C→S, and N→C. The objective decreases and then stabilizes in the displayed runs, providing a view of empirical optimization behavior across the three appearance shifts.

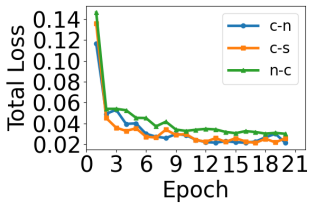


Figure 7: Training objective on three dSprites transfers.

These curves concern fitting the objective in Eq. 20, which combines squared prediction error and the empirical residual-informed term. A stable objective does not show that pseudo-label error decreases after every refinement cycle. Corollary 4 instead requires conditions on reference risk and the sensitivity of the fusion and text mappings. The plots do not measure those constants or establish that contraction condition.

E.2 Relative Branch Weights

The coefficients $\lambda_1, \lambda_2, \lambda_3$ weight the visual, text, and fusion objectives, respectively. Figure 8 displays the branch-weight sweeps on C→N. In these plots, smaller tested weights generally show faster reduction and lower final MAE, while the largest displayed value, 10, is less favorable. This observation is specific to the displayed configurations and does not establish a universal choice across domains.

The weighted-loss identity in Appendix A.5 explains why changing branch weights changes both fitting and prediction agreement in the MSE component. The full RI objective also contains the nonlinear empirical dependence term, so its optimization behavior cannot be deduced from that identity alone.

Figures 3(a)–(b) examine initialization coverage and residual-informed weighting on C→N. Their selection-risk and residual-dependence interpretations are given in Appendices A.4 and A.7; these empirical sweeps do not establish the population conditions.

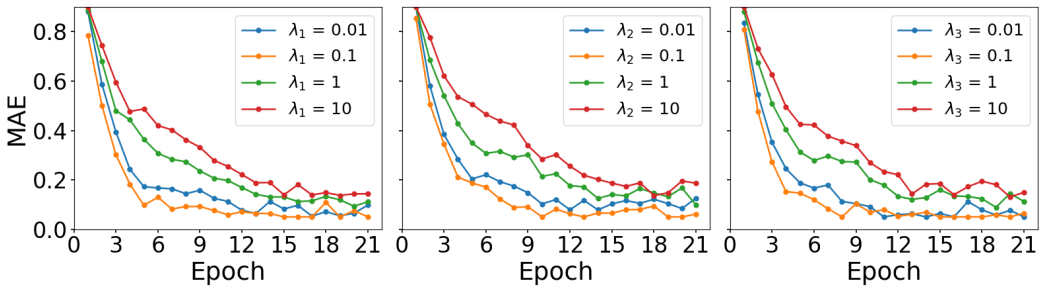


Figure 8: Branch-weight sensitivity on dSprites C→N. The three panels correspond to the visual, text, and fusion weights.

F PROMPT TEMPLATE CATALOGUE

Table 9 records the linguistic forms used to construct the prompt bank. Numerical values are supplied through the prompt constructor in Section 4.1: source labels are available for source prompts, while only prototypes or pseudo-labels supply target values. These template forms specify language variation, not a guarantee that the text encoder preserves numerical distances.

The catalogue includes geometric descriptions for dSprites, horizontal/vertical descriptions for MPI3D, and pitch/yaw/roll descriptions for BIWI. The evaluated target coordinates are those stated in Section 5.1. In particular, the orientation wording in the dSprites catalogue is not an additional evaluated regression target, and the analysis assumes no target-domain ground-truth attribute is supplied to the prompt constructor.

Table 9: Linguistic prompt forms from the supplementary template catalogue.

Prompt templates
dSprites
A shape is scaled to a specific size, rotated by a certain number of degrees, and located at a given (x, y) position.
An object has a particular scale and orientation in degrees, and is positioned horizontally at a specific x-coordinate and vertically at a specific y-coordinate.
A figure of a given size faces a certain direction in degrees and is placed at specific x and y coordinates.
An object is placed at a particular (x, y) location, with a defined scale and rotation in degrees.
A geometric shape has a relative size, is turned by a specific number of degrees, and is positioned at given x and y coordinates.
An item is scaled to a particular size, oriented at a specific angle in degrees, and located horizontally and vertically at certain positions.
A simple object is located at a specific x and y position, with a defined scale and orientation in degrees.
MPI3D
A photo shows a toy object positioned at a specific horizontal and vertical location.
An image presents a toy object located at a particular horizontal and vertical position.
A realistic photo displays a toy object placed at defined horizontal and vertical coordinates.
A candid shot captures a toy object positioned at a certain horizontal and vertical location.
A detailed picture illustrates a toy object situated at given horizontal and vertical positions.
A visual depiction shows a toy object found at a specific horizontal and vertical placement.
A clear photo presents a toy object located at particular horizontal and vertical coordinates.
Biwi Kinect Head Pose
A photo shows a person with a head pose defined by pitch, yaw, and roll angles.
A realistic image captures a person facing a certain direction, described by pitch, yaw, and roll values in degrees.
A face is seen looking in a specific direction with defined pitch, yaw, and roll angles.

Continued on next page

Table 9 (continued)

Prompt templates
A portrait depicts a person whose head is turned, with given pitch, yaw, and roll degrees.
A camera shot shows a face oriented according to pitch, yaw, and roll angles.
A head pose is described using pitch, yaw, and roll values, all expressed in degrees.
The person in the image has a head orientation characterized by specific pitch, yaw, and roll angles.