Factored Representation Learning through Mechanism Diversity

Minghao Fu[1] **Selena Ge**[1] **Hongjia Liu**[1] **Zijian Li**[2]
Fan Feng[1] **Kun Zhang**[2] **Biwei Huang**[1]

[1]University of California San Diego
[2]Carnegie Mellon University
m9fu@ucsd.edu

Abstract

A generative model is *factored* when its latent coordinates recover the underlying factors up to permutation and componentwise invertible transformations. Such representations support controllable generation and reproducible interpretation. We study conditional generators $\mathbf{x} = g(\mathbf{z}, \mathbf{u})$ whose mechanisms vary with $\mathbf{u}$ while the latent prior remains fixed. We prove exact componentwise identifiability from unpaired conditional distributions when factor response profiles are diverse, response directions are shared across conditions, and the reference tangent-space map is injective. The result permits dependent latent coordinates and nonparametric latent densities. Motivated by this mechanism-diversity perspective, we introduce *Mechanistic Contrastive Learning* (MCL), a model-agnostic objective that contrasts critic-derived mechanism gradients with conditioning signals. Empirically, our conditional diffusion pipeline achieves strong latent concept disentanglement on three benchmarks, and MCL improves visual prediction in matched world-model comparisons built on a 1.4B video backbone, including real-robot data.

Code

1 Introduction

What makes a generative model *factored*? Intuitively, a factored model has a latent space where each coordinate controls one factor of variation: one latent for color, another for shape, a third for viewpoint, and so on. Such representations underpin controllable generation [1], compositional generalization [2], and reproducible science [3]. Yet whether the latent spaces of modern conditional generative models are provably factored remains an open question.

Modern conditional generative models share a common architecture: a conditioning signal $\mathbf{u}$, such as a text prompt, label, or action, modulates a generator $g(\cdot, \mathbf{u})$ mapping latents $\mathbf{z}$ to observations $\mathbf{x}$ [1, 4, 5]. Prior work explains part of their empirical advantage through simpler conditional subproblems [6]; here we ask what this conditioning structure implies for identifiability. A prominent line of nonlinear-ICA theory varies $p(\mathbf{z} \mid \mathbf{u})$ while keeping the decoder fixed [7, 8], a regime we call *Latent Representation Diversity* (LRD). We instead study *Generating Mechanism Diversity* (GMD), where conditioning reshapes the generator at a fixed latent prior. Our contributions are threefold. ***(i) Theory.*** We establish exact componentwise identifiability from unpaired conditional laws under diverse generator response profiles, shared response directions, and geometric observability (Theorem 1), without imposing latent-coordinate independence. A separate model-class comparison specifies when conditional generation has greater representational capacity than a fixed-decoder family (Proposition 1). ***(ii) Method.*** MCL contrasts critic-derived mechanism gradients with conditioning embeddings using shuffled conditions as negatives. We analyze projected mechanism-gradient

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

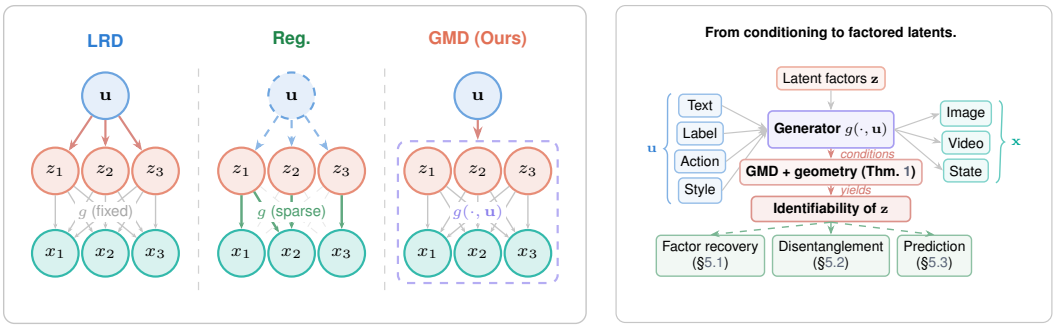


Figure 1: **Factored generative models through mechanism diversity. Left:** LRD varies $p(\mathbf{z}|\mathbf{u})$, **Reg.** constrains g, and GMD (ours) varies $g(\cdot, \mathbf{u})$. **Right:** the theorem combines mechanism diversity with geometric conditions; dashed links indicate empirical evaluation themes.

diversity under a fixed-key contrastive objective. ***(iii) Empirics.*** We evaluate factor recovery with MD-VAE on controlled synthetic data and with MDDiff on three image benchmarks. Matched Wan 1.4B comparisons evaluate MCL's effect on Robotwin prediction and transfer, and on DROID real-robot prediction.

2 Background and Related Work

From Nonlinear ICA to Factored Generative Models. Nonlinear ICA studies when a nonlinear mixing of latent factors can be inverted uniquely up to componentwise transformations and permutation. Identifiability matters because fitting the observation distribution alone leaves the latent coordinates ambiguous [9]. Conditional generative modeling supplies additional distributions from which to resolve this ambiguity under suitable structure.

Three principles for factored representations. Existing answers to nonlinear ICA and generative-model identifiability largely follow these two routes: **LRD**, which varies the latent distribution sufficiently, and **regularization**, which constrains the model's degree of freedom. ***Beyond these***, we identify **GMD** as a third principle, varying the generator itself through conditioning.

(1) Latent distribution diversity (LRD). The auxiliary $\mathbf{u}$ creates diversity in the latent distribution $p(\mathbf{z} \mid \mathbf{u})$ while the generator g stays fixed. Hyvarinen and Morioka [10] established this for temporal nonstationarity, Hyvarinen et al. [7] extended it to arbitrary auxiliaries through generalized contrastive estimation, and Khemakhem et al. [8] unified the framework with VAEs by showing that the ELBO with an auxiliary-conditioned prior recovers identifiable representations. Subsequent work studied partial and temporal identifiability using log-density derivatives [11, 12] and extended LRD identifiability to multi-view [13], partially observed [14], and additive noisy [15] settings. Apart from theorems, several practical methods exploit LRD implicitly: contrastive predictive coding leverages temporal structure [16, 17], and Zimmermann et al. [18] showed that it can invert the data-generating process to yield identifiable features. Reizinger et al. [19] proved that cross-entropy classification yields identifiable representations, and Reizinger et al. [20] demonstrated that policy diversity produces identifiable state representations in reinforcement learning. The shared technical formal condition underlying these results is summarized below, and our GMD condition (Theorem 1) replaces the requirement on the latent log-density q_i with a condition on the generator Jacobian.

Assumption 1 (Sufficient LRD [7])

For any $\mathbf{z} \in \mathbb{R}^{d_z}$, there exist $2d_z+1$ values $\mathbf{u}^{(0)}, \ldots, \mathbf{u}^{(2d_z)}$ such that the $2d_z$ vectors $\mathbf{w}(\mathbf{z}, \mathbf{u}^{(j)}) - \mathbf{w}(\mathbf{z}, \mathbf{u}^{(0)})$, $j = 1, \ldots, 2d_z$, are linearly independent, where $\mathbf{w}(\mathbf{z}, \mathbf{u}) = \left(\frac{\partial q_i(z_i, \mathbf{u})}{\partial z_i}, \frac{\partial^2 q_i(z_i, \mathbf{u})}{\partial z_i^2}\right)_{i=1}^{d_z}$ and q_i is the log-conditional density of z_i.

(2) Regularization. Instead of requiring diversity in the data, this family constrains the model architecture or objective to favor factored solutions. Independence-based methods penalize statistical dependence among latents: β-VAE [21] upweights the KL term, while FactorVAE [22] and β-TCVAE [23] directly minimize total correlation. Sparsity-based methods constrain the generator: Lachapelle et al. [24] penalize off-diagonal Jacobian entries (mechanism sparsity), Zheng et al. [25] prove identifiability under sparsity on g without auxiliary variables, and Zhang et al. [26]

assume a causal graph among latents across distributions. Compression-based methods restrict the latent bottleneck: Tripod [27] combines finite scalar quantization, kernel-based multiinformation minimization, and a normalized Hessian penalty, showing that three complementary regularizers together achieve strong disentanglement in autoencoders. Locatello et al. [2] showed that without such inductive biases, unsupervised disentanglement is impossible.

(3) GMD (ours). GMD places diversity in the generator: a conditioning signal $\mathbf{u}$ reshapes $g(\cdot, \mathbf{u})$ while $p(\mathbf{z})$ stays fixed. Conditional GANs [5, 28], VAEs [4, 29], diffusion models [1, 30, 31], and action-conditioned world models [32–34] already use this architectural pattern. Our contribution is to characterize when variation across these mechanisms identifies a common system of latent factors. The data-partitioning view [6] explains conditional generation's approximation advantage through simpler conditional subproblems; GMD asks the complementary question of when conditional distributions determine the factors themselves. Representational capacity and identifiability are distinct: Proposition 1 compares the generating families, whereas Theorem 1 establishes componentwise identification within an explicitly stated model class. Unlike fixed-decoder LRD, its diversity condition concerns generator response profiles and permits dependent latent coordinates. Shared response directions and geometric observability allow unpaired conditional distributions to determine one correspondence across conditions; diversity then resolves the individual components. MCL (Section 4) turns mechanism variation into a practical training objective, connecting this identifiability perspective to conditional generative modeling.

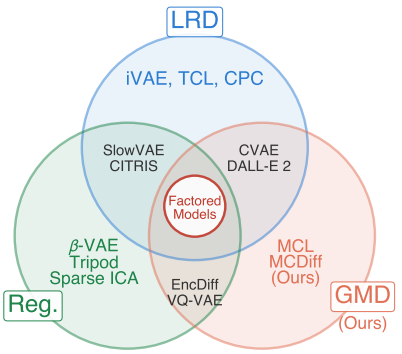


Figure 2: **Three principles for factored models. LRD**: diversify the latent distribution. **Regularization**: constrain the model (sparsity, quantization, Hessian). **GMD** (ours): diversify the generator via conditioning. Capacity comparison under the model-class conditions of Proposition 1.

3 When Is a Generative Model Factored?

We distinguish representational capacity from identifiability. Section 3.1 gives a capacity criterion for specified conditional model classes. Section 3.2 then proves exact componentwise identification under diverse response profiles and geometric conditions. Section 3.3 describes a temporal specialization and explains the additional information required by paired-view arguments.

3.1 Capacity Comparison of GMD and LRD

Consider two data-generation families over $\mathbf{u} \in \mathcal{U}$, $\mathbf{z} \in \mathcal{Z}$, and $\mathbf{x} \in \mathcal{X}$, with source noise $\boldsymbol{\epsilon} \sim P_{\boldsymbol{\mathcal{E}}}$:

$$\underbrace{\mathbf{z} = f_L(\mathbf{u}, \boldsymbol{\epsilon}), \quad \mathbf{x} = g_L(\mathbf{z}),}_{\text{LRD: } \mathbf{u} \text{ shifts } p(\mathbf{z}|\mathbf{u}),\ g \text{ fixed}} \qquad \underbrace{\mathbf{z} = f_G(\boldsymbol{\epsilon}), \quad \mathbf{x} = g_G(\mathbf{u}, \mathbf{z})}_{\text{GMD: } \mathbf{u} \text{ reshapes } g,\ p(\mathbf{z}) \text{ fixed}}.$$

Let $\mathcal{P}_{\mathrm{LRD}}$ and $\mathcal{P}_{\mathrm{GMD}}$ denote the conditional-law families induced by two specified model classes. Their relative capacity depends on these classes; the following criterion establishes when GMD has strictly greater capacity.

Proposition 1 (Capacity separation under an observable-rank gap)**.** *Suppose $\mathcal{P}_{\mathrm{LRD}} \subseteq \mathcal{P}_{\mathrm{GMD}}$. Let $\Psi_L : \Theta_L \to \mathbb{R}^q$ and $\Psi_G : \Theta_G \to \mathbb{R}^q$ be real-analytic maps giving the same finite collection of observable statistics of the conditional laws, on open parameter domains. If, at some θ_G,*

$$\operatorname{rank} D\Psi_G(\theta_G) > \sup_{\theta_L \in \Theta_L} \operatorname{rank} D\Psi_L(\theta_L),$$

then $\mathcal{P}_{\mathrm{LRD}} \subsetneq \mathcal{P}_{\mathrm{GMD}}$.

Proposition 1 compares the two model classes through the ranks of their parameter-to-statistic maps. For unrestricted conditional latent laws and sufficient latent dimension, a fixed decoder can reproduce arbitrary conditional observation laws. Proposition 1 therefore separates capacity under an explicit criterion; the identifiability result below is a distinct statement within its own geometric model class.

3.2 Identifiability through Mechanism Diversity

We first formalize what it means for a generative model to have a factored latent space.

Definition 1 (Factored Generative Model)**.** *A model $\mathbf{x} = g(\mathbf{z}, \mathbf{u})$ is* factored within a model class *if every observationally equivalent model in that class satisfies $\hat{z}_i = h_i(z_{\pi(i)})$ for one permutation π and scalar invertible functions $\{h_i\}_{i=1}^{d_z}$ shared across all conditions.*

This is the componentwise identifiability notion of nonlinear ICA [7, 8], with an explicit requirement that the same latent correspondence work across conditions. We use *factored* and *identifiable up to permutation and componentwise transformations* interchangeably within the stated model class.

Let $\mathcal{U} = \{\mathbf{u}^{(0)}, \ldots, \mathbf{u}^{(m)}\}$ be ordinary condition labels and write $g_0 = g_{\mathbf{u}^{(0)}}$. We use the full Jacobian $Dg_{\mathbf{u}}$, allowing $d_x > d_z$, and define the reference tangent-space map $\Gamma(\mathbf{z}) = \operatorname{col} Dg_0(\mathbf{z})$.

Assumption 2 (Sufficient GMD)

The family has shared factor response directions:

$$\partial_i g_{\mathbf{u}}(\mathbf{z}) = a_i(\mathbf{z}, \mathbf{u})\, \partial_i g_0(\mathbf{z}), \qquad a_i(\mathbf{z}, \mathbf{u}) \neq 0, \quad a_i(\mathbf{z}, \mathbf{u}^{(0)}) = 1.$$

Its mechanism profiles $\mathbf{a}_i(\mathbf{z}) = (a_i(\mathbf{z}, \mathbf{u}))_{\mathbf{u} \in \mathcal{U}}$ satisfy

$$\operatorname{rank}[\mathbf{a}_i(\mathbf{z})\ \mathbf{a}_j(\mathbf{z})] = 2 \qquad (i \neq j)$$

on a dense subset of $\mathcal{Z}$. The gains a_i may depend on all latent coordinates.

Theorem 1 (Identifiability under GMD)

Let $\mathcal{Z}, \widehat{\mathcal{Z}}$ be products of open intervals, with positive C^1 latent densities independent of $\mathbf{u}$. Let $g_{\mathbf{u}}, \hat{g}_{\mathbf{u}}$ be proper C^2 embeddings into $\mathbb{R}^{d_x}$, $d_x > d_z$. Suppose g satisfies Assumption 2, $\hat{g}$ satisfies its shared-direction condition, and Γ is injective. If $(g_{\mathbf{u}})_{\#} p_{\mathbf{z}} = (\hat{g}_{\mathbf{u}})_{\#} p_{\hat{\mathbf{z}}}$ for every $\mathbf{u}$, then

$$\hat{g}_{\mathbf{u}}^{-1} \circ g_{\mathbf{u}} = H, \qquad H_i(\mathbf{z}) = h_i(z_{\pi(i)}) \quad \text{for every } \mathbf{u},$$

for one permutation π and scalar C^2 diffeomorphisms h_i. Inverses are taken on the embedded images; no paired observations are required.

Scope of the assumptions. Diversity separates factor response profiles; two conditions can suffice. Shared response directions and an injective tangent-space map provide the geometric structure used to align latent states across conditions. They allow interacting nonlinear generators but exclude unrestricted $g(\mathbf{z}, \mathbf{u})$; no independence of latent components is required. The theorem concerns noiseless embedded manifolds, whose tangent spaces distinguish latent states.

Proof sketch. Equality of unpaired laws identifies each embedded image and its tangent spaces. Shared directions imply $\Gamma(r_{\mathbf{u}}(\hat{\mathbf{z}})) = \widehat{\Gamma}(\hat{\mathbf{z}})$ for $r_{\mathbf{u}} = g_{\mathbf{u}}^{-1} \circ \hat{g}_{\mathbf{u}}$; injectivity of Γ therefore proves $r_{\mathbf{u}} = r$ for all $\mathbf{u}$. Differentiating gives $D_{\mathbf{u}} Dr = Dr \widehat{D}_{\mathbf{u}}$, where $D_{\mathbf{u}}$ and $\widehat{D}_{\mathbf{u}}$ are diagonal gain matrices. Pairwise independent mechanism profiles force Dr to have one nonzero entry per row and column. Connected product domains then yield a single permutation and scalar diffeomorphisms. Appendix C.2 gives the full proof.

3.3 Temporal Contexts and the Scope of Unpaired Conditioning

An observed condition need not be an externally supplied label: it can be a fixed function of past observations. Proposition 2 applies Theorem 1 to exogenous next-step innovations under a fixed, shared context map. The formal statement and proof are in Appendices B.1 and C.3.

Proposition 2 (GMD with an observed temporal context; *informal*)**.** *Fix a context map $U_t = e(\mathbf{x}_{\leq t})$. If the next-step innovation is independent of the past and both models, compared using this same context map, satisfy all conditions of Theorem 1, then the innovation is identifiable up to one permutation and componentwise transformations shared across contexts.*

Unpaired conditions and multi-view data. Our observations specify separate conditional laws $p(\mathbf{x} \mid \mathbf{u})$. Paired multi-view data additionally specify the coupling between observations of the same

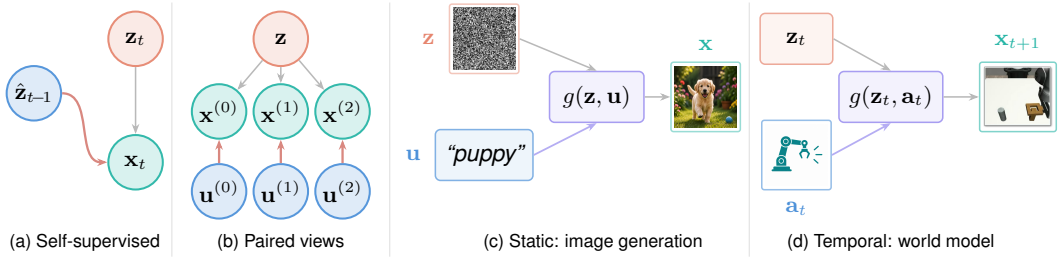


Figure 3: **Conditioning regimes and instantiations of GMD.** (a) Temporal context: a fixed representation $\hat{\mathbf{z}}_{t-1}$ of past observations acts as $\mathbf{u}$. (b) Paired multi-view data: several observations share one latent instance; their joint distribution contains information absent from unpaired conditional laws. (c) Static GMD: a text prompt $\mathbf{u}$ reshapes the denoiser. (d) Temporal GMD: an action $\mathbf{a}_t$ reshapes the dynamics at the pixel level.

latent instance. Multi-view identification arguments use paired joint observations and additional assumptions [35, 36], a different observation setting illustrated in Figure 3b. Extending the present unpaired theorem to noisy or non-injective generators remains open; Appendix C.4 explains this distinction.

4 Learning with Mechanism Contrast

A conditional architecture alone does not ensure that different conditioning values induce distinguishable mechanisms. We first describe two generative instantiations, then introduce MCL, which contrasts mechanism-critic gradients across conditioning signals. This practical objective is motivated by mechanism diversity.

4.1 GMD Instantiations

Static case: conditional generative models. For controlled synthetic evaluation (Section 5.1) we construct a *Mechanism-Diverse VAE* (**MD-VAE**): a conditional VAE with encoder $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{u})$ and decoder $p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u})$, fixed prior $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$, trained by maximizing the conditional ELBO

$$\log p_\theta(\mathbf{x} \mid \mathbf{u}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{u})}[\log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u})] - \mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{u}) \,\|\, p(\mathbf{z}))\,, \tag{1}$$

denoted $\mathcal{L}_{\text{MD-VAE}}$. Maximizing it reshapes the decoder $g(\cdot, \mathbf{u})$ with $\mathbf{u}$ at fixed prior, instantiating GMD by construction. In the spirit of latent diffusion [1], where a standard VAE encoder feeds a diffusion model in latent space, our **MDDiff** (Section 5.2) plugs MCL into the same role and runs $\mathbf{u}$-conditioned diffusion on the resulting mechanism-diverse latents.

Temporal case: state-space models. The conditioning signal is naturally provided by a history:

$$\mathbf{x}_t = g(\mathbf{x}_{t-H:t-1},\, \mathbf{u}_{t-H:t-1})\,,$$

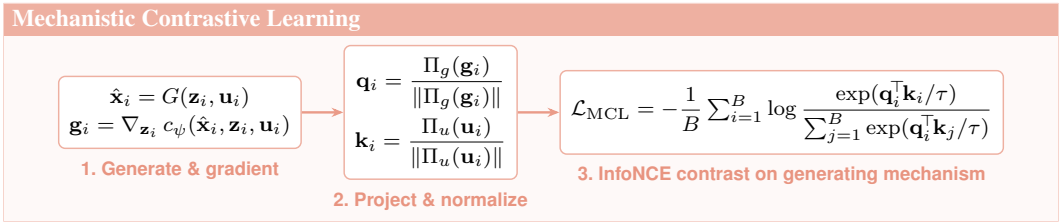
where $\mathbf{u}_{t-H:t-1}$ can be actions in action-conditioned world models [33, 37], learned latent states in the self-supervised learning, *e.g.*, time-series data, or any domain index. Temporal conditioning provides a practical source of mechanism variation; identifying a latent state additionally requires the model-class conditions of Theorem 1.

4.2 Mechanistic Contrastive Learning

We introduce a model-agnostic objective that encourages condition-discriminative mechanism-gradient features. Appendix B.2 characterizes projected-feature diversity in an idealized fixed-key setting; Appendix C.6 distinguishes that result from the generator-level conditions required for identifiability.

MCL augments any differentiable conditional generator G with three lightweight modules trained jointly with it: a *mechanism critic* $c_\psi : \mathcal{X} \times \mathcal{Z} \times \mathcal{U} \to \mathbb{R}$ that scores how strongly the realized mechanism $G(\cdot, \mathbf{u})$ depends on $\mathbf{z}$, and two L_2-normalized projection heads Π_g, Π_u mapping

mechanism-gradient features and conditioning features into a shared embedding space. For each in-batch sample $(\mathbf{z}_i, \mathbf{u}_i)$ we generate $\hat{\mathbf{x}}_i = G(\mathbf{z}_i, \mathbf{u}_i)$, score it with the critic, and read off the *mechanism gradient* $\mathbf{g}_i = \nabla_{\mathbf{z}_i} c_\psi(\hat{\mathbf{x}}_i, \mathbf{z}_i, \mathbf{u}_i)$, i.e. the gradient of the critic through the realized generator. The gradient and conditioning embeddings are then contrasted with InfoNCE. The three stages are shown below.



The full training loss is $\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{MCL}}\mathcal{L}_{\mathrm{MCL}}$, where $\mathcal{L}_{\mathrm{rec}}$ is the host generator's native reconstruction objective and $\lambda_{\mathrm{MCL}} > 0$ controls the mechanism-contrastive term. MCL favors agreement between a mechanism-gradient feature and its own conditioning embedding relative to shuffled conditions. Theorem 2 characterizes projected-feature diversity in the fixed-key setting; Section 5 evaluates MCL's effect on factor recovery and prediction quality.

Comparison with contrastive learning and L_1/L_2 regularization. MCL differs from two commonly used families of objectives (Figure 4). Some ***contrastive-learning*** methods exploit auxiliary distribution shifts through predictive coding [16, 17], nonlinear-ICA objectives [7, 18], or classification on auxiliary labels [19]. Their contrastive signal acts on learned representations. ***Regularization*** instead favors structural properties through objectives such as sparsity on the generator Jacobian [24, 25], quantization [38, 39], or the combined quantization, mutual-information, and Hessian penalties of Tripod [27]. ***MCL*** contrasts critic gradients through the generator against the conditioning signal. The fixed generative prior $p(\mathbf{z})$ and the conditioning-dependent generator $g(\mathbf{z}, \mathbf{u})$ remain the host model's architecture; MCL adds a mechanism-gradient training signal without requiring a conditional prior.

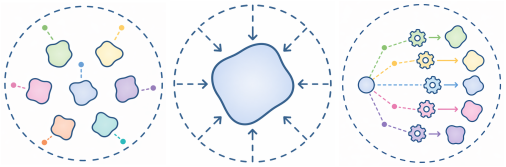


Figure 4: **What each family of objectives does in latent space.** (Left) **Contrastive learning**, LRD; (Center) L_1/L_2 **regularization**; (Right) **MCL**.

5 Experiments

We validate the theory on synthetic data with known ground truth (§5.1), then evaluate concept disentanglement (static case, §5.2) and world modeling (temporal case, §5.3).

5.1 Experiments on Synthetic Data for Factor Verification

Setup and metric. We generate data from a known GMD process (full data generation pipeline in Appendix D.3) with $d_z \in \{5, 10, 20\}$ and a default of $|\mathcal{U}| = 8$ conditioning values. We compare three baselines drawn from the three principles of Section 2: (i) **β-VAE** [21] (regularization-based), (ii) **iVAE** [8] (LRD-based), and (iii) **MD-VAE** (ours, GMD-based), trained with the conditional ELBO of Equation (1). Identifiability is measured by the Mean Correlation Coefficient (MCC) between true and recovered latent variables.

Results. Across $d_z \in \{5, 10, 20\}$, MD-VAE attains median MCC $\{0.96, 0.87, 0.81\}$, compared with ranges 0.80 to 0.65 for iVAE and 0.53 to 0.39 for β-VAE. These results favor conditional decoding in the evaluated setting. Appendix F.1 studies the effect of conditioning-set size.

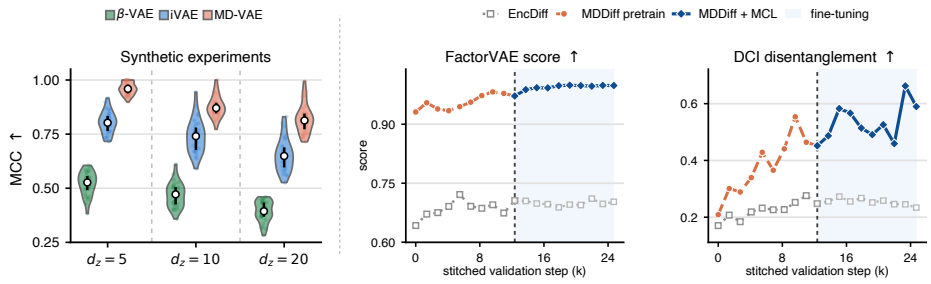


Figure 5: **Synthetic experiments and `Cars3D` metrics curves during training.** Left: MCC violin plots over 30 seeds per cell. Right: stitched FactorVAE and DCI trajectories on `Cars3D`.

5.2 Experiments on Concept Disentanglement

As a representative static GMD task, we evaluate whether MCL improves concept disentanglement in conditional diffusion models (concept tokens $= \mathbf{u}$, denoiser $= g(\cdot, \mathbf{u})$). Our **MDDiff** builds MCL and mechanism diversity on top of a latent diffusion generative model with cross-attention, following Yang et al. [40]; full architecture and training details are in Appendix D.

Datasets. We use three standard disentanglement benchmarks: `Shapes3D` [22], `MPI3D` [41], and `Cars3D` [42], which together cover synthetic factors, 3D object factors, and viewpoint variation. All experiments use 64×64 images, following [22, 23, 43, 44].

Baselines and metrics. We compare VAE baselines (FactorVAE [22], β-TCVAE [23]), GAN baselines (InfoGAN-CR [45], GANSpace [46], LatentDisco [47], DisCo [43]), and diffusion baselines (DisDiff [44], EncDiff [40]) against MDDiff; full descriptions are in Appendix D.2. All methods use $N = 20$ scalar representations [44]; we report mean $\pm$ std over 15 runs using the FactorVAE score and DCI disentanglement (as in Appendix D.1).

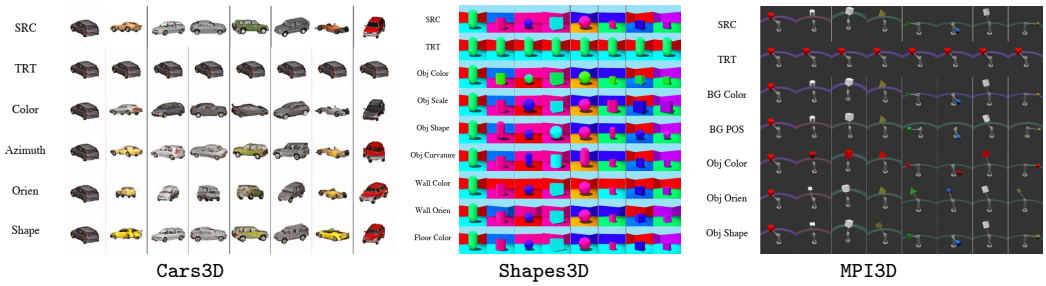


Figure 6: **Factor swapping across all three benchmarks.** Row 1: source images (SRC); row 2: target images (TRT); subsequent rows: SRC with one latent dimension replaced by the corresponding TRT dimension, we match the concept through their consistent visual variations across samples.

Results and analysis. Table 1 compares the full pipelines. MDDiff has the highest mean FV score on all three benchmarks and DCI on `Cars3D` and `Shapes3D`; EncDiff has the highest DCI on `MPI3D`. The largest DCI difference is on `Cars3D`, 0.493 versus 0.279 for EncDiff. Factor swapping (Figure 6) and latent traversals (Appendix F) provide qualitative evidence of localized attribute control. Generation-quality comparisons are reported in Appendix F.2.

5.3 Experiments on World Modeling

We evaluate whether MCL improves visual prediction in action-conditioned world models. These experiments assess prediction quality under recorded or inferred actions.

Forward and inverse dynamics. Video-action models recover actions from observed transitions [48, 49], while our forward model generates future observations from a state representation and action: $\mathbf{o}_{t+1} = g_{\text{fwd}}(\mathbf{z}_t, \mathbf{a}_t)$. Actions therefore provide the conditioning signal $\mathbf{u}_t = \mathbf{a}_t$. An inverse-dynamics model can supply latent actions when recorded actions are unavailable. This places the architecture in the conditional-generation regime.

Table 1: **Disentanglement comparison** (mean ± std; best results are in **Bold**).

Type	Method	Cars3D		Shapes3D		MPI3D	
		FV score ↑	DCI ↑	FV score ↑	DCI ↑	FV score ↑	DCI ↑
VAE	FactorVAE	0.906± 0.052	0.161± 0.019	0.840± 0.066	0.611± 0.082	0.152± 0.025	0.240± 0.051
	β-TCVAE	0.855± 0.082	0.140± 0.019	0.873± 0.074	0.613± 0.114	0.179± 0.017	0.237± 0.056
GAN	InfoGAN-CR	0.411± 0.013	0.020± 0.011	0.587± 0.058	0.478± 0.055	0.439± 0.061	0.241± 0.075
	LatentDisco	0.852± 0.039	0.216± 0.072	0.805± 0.064	0.380± 0.062	0.391± 0.039	0.196± 0.038
	GANSpace	0.932± 0.018	0.209± 0.031	0.788± 0.091	0.284± 0.034	0.465± 0.036	0.229± 0.042
	DisCo	0.855± 0.074	0.271± 0.037	0.877± 0.031	0.708± 0.048	0.371± 0.030	0.292± 0.024
Diffusion	DisDiff	0.976± 0.018	0.232± 0.019	0.902± 0.043	0.723± 0.013	0.617± 0.070	0.337± 0.057
	EncDiff	0.773± 0.060	0.279± 0.022	0.999± 0.000	0.969± 0.030	0.872± 0.049	**0.685**± 0.044
	MDDiff (Ours)	**0.985**± 0.048	**0.493**± 0.025	**1.000**± 0.000	**0.993**± 0.030	**0.930**± 0.049	0.679± 0.010

Architecture. Our action-conditioned generative world model (ACG) instantiates the forward SEM above on top of **Wan 1.4B** [50], a general-purpose large-scale pretrained video generation backbone, which we adapt to embodied tasks. We instantiate two regimes of ACG that differ in whether the action conditioning is observed or inferred: **Wan-ACG** feeds the recorded action $\mathbf{a}_t$ directly to the FDM, so the conditioning $\mathbf{u}_t = \mathbf{a}_t$ is observed; **Wan-LACG** (the L denotes *latent* action) replaces the recorded action with an IDM-inferred latent action $\mathbf{u}_t$. Across both regimes we connect Wan 1.4B with three components: (i) an inverse dynamics model (IDM) that infers a diagonal-Gaussian posterior over a d_z-dimensional latent action for every transition (used in Wan-LACG only); (ii) an action interface A_ψ that lifts the frame-rate action sequence into FDM-aligned conditioning through AdaLN modulation; and (iii) a forward dynamics model (FDM), initialized from the pretrained Wan 1.4B backbone, that performs flow matching to predict future latent video dynamics.

MCL on the action conditioning. MCL attaches to the action channel of the FDM: a mechanism critic c_ψ produces the gradient $\mathbf{g}_t = \nabla_{\mathbf{z}_t} c_\psi(\hat{\mathbf{x}}_{t+1}, \mathbf{z}_t, \mathbf{u}_t)$, where $\mathbf{z}_t$ is the latent state (the encoder embedding of $\mathbf{o}_t$) and $\mathbf{u}_t$ is the action conditioning ($\mathbf{a}_t$ in Wan-ACG, the IDM-inferred latent action in Wan-LACG); $\mathbf{g}_t$ is contrasted via InfoNCE against the projected action embedding. The IDM, action interface, FDM, and MCL projection heads are trained jointly with the causal VAE kept frozen, under $\mathcal{L}_{\text{rec}} + \beta\,\mathcal{L}_{\text{KL}} + \lambda_{\text{adv}}\,\mathcal{L}_{\text{GRL}} + \lambda_{\text{MCL}}\,\mathcal{L}_{\text{MCL}}$. Full hyperparameters are in Appendix D.

Protocol and methods. We evaluate on Robotwin [51] under the low-data target-embodiment setting (Table 2), reporting SSIM, PSNR, MSE, and last-frame SSIM-L on the target task (`place_a2b_left`) and the zero-shot transfer task (`place_a2b_right`). For each of the two regimes (Wan-ACG, Wan-LACG) we report the variant without MCL (only the host objective) and with MCL added; the comparison tests visual prediction quality against the matching no-MCL baseline in each regime.

Table 2: **World-model evaluation on Robotwin under the low-data target-embodiment setting. Target task**: held-out evaluation on `place_a2b_left` (same task as training). **Transfer task**: zero-shot evaluation on `place_a2b_right`. Higher is better for SSIM, PSNR, and SSIM-L; lower is better for MSE. **Bold** marks the best result per column.

Method	Target task				Transfer task			
	SSIM ↑	PSNR ↑	MSE ↓	SSIM-L ↑	SSIM ↑	PSNR ↑	MSE ↓	SSIM-L ↑
Wan-ACG	0.763	18.61	0.0138	0.735	0.763	18.45	0.0143	0.732
Wan-ACG+MCL	**0.765**	**18.73**	**0.0134**	**0.737**	**0.765**	**18.58**	**0.0138**	**0.739**
Wan-LACG	0.769	18.98	0.0127	0.744	0.782	19.31	0.0117	0.752
Wan-LACG+MCL	**0.772**	**18.99**	**0.0125**	**0.749**	**0.784**	**19.33**	**0.0115**	**0.758**

Real-robot evaluation on DROID. We further evaluate on DROID [52], using the held-out split of Ctrl-World [53] and the 14-step autoregressive protocol of PersistWorld [54], generating 70 frames per rollout. Starting from the same Wan 1.4B checkpoint, we post-train Wan-ACG with and without MCL while holding the data, action interface, update count, evaluation clips, and sampling settings fixed. Table 3 reports external-camera prediction quality: MCL improves SSIM by 0.009 and PSNR

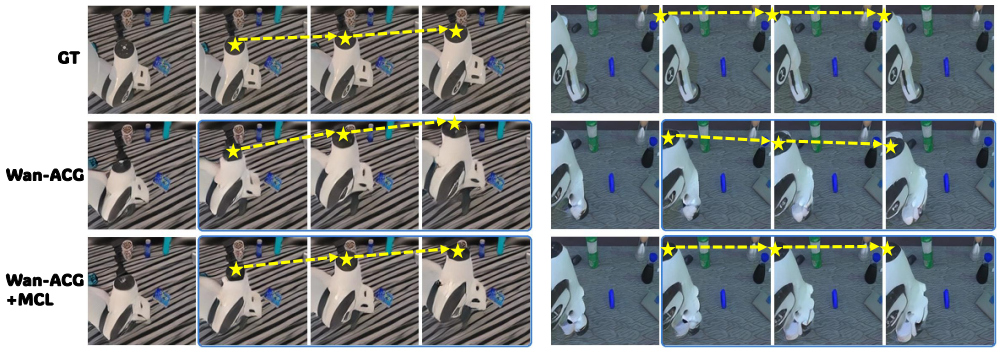


Figure 7: **World-model qualitative ablation.** Two manipulation rollouts driven by the same initial observation and ground-truth action sequence. **Top**: ground-truth frames (**GT**). **Middle**: Wan-ACG alone. **Bottom**: Full method **Wan-ACG+MCL**. The dashed yellow trajectory traces the gripper end-effector across frames.

by 0.35 dB, and reduces LPIPS by 0.006. These results extend the visual-prediction evaluation to real-robot data.

Table 3: **Real-robot prediction on DROID.** Matched Wan 1.4B variants evaluated on external-camera rollouts.

Method	SSIM ↑	PSNR ↑	LPIPS ↓
Wan-ACG	0.842	23.81	0.084
Wan-ACG+MCL	**0.851**	**24.16**	**0.078**

6 Conclusion

Generator response diversity can identify a common set of latent components from unpaired conditional laws within a geometrically specified model class. Our theorem separates the alignment of latent coordinates across conditions from the resolution of individual factors. MCL translates the mechanism-diversity perspective into a practical training signal, with feature-level theory and empirical evaluations on image representations and robotic prediction.

Limitations and future work. The identifiability guarantee assumes noiseless embedded manifolds, shared factor response directions, and an injective tangent-space map. Establishing these geometric conditions for trained neural generators and connecting them to practical MCL are further theoretical questions. Appendix C.6 analyzes the critic's direct input dependencies. The image benchmarks test labeled factors, while the world-model metrics assess prediction fidelity. Extending the theory to noisy data, learned contexts, and broader conditional architectures, and connecting the practical objective to generator-level assumptions, remain open directions (Appendix H).

Acknowledgments and Disclosure of Funding

The authors thank the anonymous reviewers for their helpful comments and suggestions during the reviewing process. The authors acknowledge support from NSF DMS-2428058. The authors declare no competing interests.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[2] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning

of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR, 2019.

[3] Geoffrey Roeder, Luke Metz, and Durk P Kingma. On linear identifiability of learned representations. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[4] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[5] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[6] Fan Bao, Chongxuan Li, Jiacheng Sun, and Jun Zhu. Why are conditional generative models better than unconditional ones? In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[7] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd international conference on artificial intelligence and statistics*, pages 859–868. PMLR, 2019.

[8] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pages 2207–2217. PMLR, 2020.

[9] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[10] Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial intelligence and statistics*, pages 460–469. PMLR, 2017.

[11] Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial disentanglement for domain adaptation. In *International conference on machine learning*, pages 11455–11472. PMLR, 2022.

[12] Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. *Advances in Neural Information Processing Systems*, 35:26492–26503, 2022.

[13] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34: 16451–16467, 2021.

[14] Dingling Yao, Danru Xu, Sébastien Lachapelle, Sara Magliacane, Perouz Taslakian, Georg Martius, Julius von Kügelgen, and Francesco Locatello. Multi-view causal representation learning with partial observability. *arXiv preprint arXiv:2311.04056*, 2023.

[15] Hermanni Hälvä, Sylvain Le Corff, Luc Lehéricy, Jonathan So, Yongjie Zhu, Elisabeth Gassiat, and Aapo Hyvarinen. Disentangling identifiable features from noisy data with structured nonlinear ica. *Advances in Neural Information Processing Systems*, 34:1624–1633, 2021.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[17] Alicja Ziarko, Michał Bortkiewicz, Michał Zawalski, Benjamin Eysenbach, and Piotr Miłoś. Contrastive representations for temporal reasoning. *Advances in Neural Information Processing Systems*, 38:109229–109259, 2026.

[18] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International conference on machine learning*, pages 12979–12990. PMLR, 2021.

[19] Patrik Reizinger, Alice Bizeul, Attila Juhos, Julia E Vogt, Randall Balestriero, Wieland Brendel, and David Klindt. Cross-entropy is all you need to invert the data generating process. *arXiv preprint arXiv:2410.21869*, 2024.

[20] Patrik Reizinger, Balint Mucsanyi, Siyuan Guo, Benjamin Eysenbach, Bernhard Schölkopf, and Wieland Brendel. Skill learning via policy diversity yields identifiable representations for reinforcement learning. *arXiv preprint arXiv:2507.14748*, 2025.

[21] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[22] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pages 2649–2658. PMLR, 2018.

[23] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in neural information processing systems*, 2018.

[24] Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. In *Conference on Causal Learning and Reasoning*, pages 428–484. PMLR, 2022.

[25] Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *Advances in neural information processing systems*, 35:16411–16422, 2022.

[26] Kun Zhang, Shaoan Xie, Ignavier Ng, and Yujia Zheng. Causal representation learning from multiple distributions: A general setting. *arXiv preprint arXiv:2402.05052*, 2024.

[27] Kyle Hsu, Jubayer Ibn Hamid, Kaylee Burns, Chelsea Finn, and Jiajun Wu. Tripod: Three complementary inductive biases for disentangled representation learning. In *International Conference on Machine Learning*, 2024.

[28] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–842, 2021.

[29] William Harvey, Saeid Naderiparizi, and Frank Wood. Conditional image generation by conditioning variational auto-encoders. *arXiv preprint arXiv:2102.12037*, 2021.

[30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[31] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[32] Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6650–6656. IEEE, 2021.

[33] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[34] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025.

[35] Yingyao Hu and Susanne M Schennach. Instrumental variable treatment of nonclassical measurement error models. *Econometrica*, 76(1):195–216, 2008.

[36] Minghao Fu, Biwei Huang, Zijian Li, Yujia Zheng, Ignavier Ng, Guangyi Chen, Yingyao Hu, and Kun Zhang. Learning general causal structures with hidden dynamic process for climate analysis. 2025.

[37] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[38] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

[39] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, 2019.

[40] Tao Yang, Cuiling Lan, Yan Lu, and Nanning Zheng. Diffusion model with cross attention as an inductive bias for disentanglement. *Advances in Neural Information Processing Systems*, 37: 82465–82492, 2024.

[41] 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. In *Advances in Neural Information Processing Systems*, 2019.

[42] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in neural information processing systems*, 2015.

[43] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2021.

[44] Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2023.

[45] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International conference on machine learning*, pages 6127–6139. PMLR, 2020.

[46] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. In *Advances in neural information processing systems*, 2020.

[47] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In *International conference on machine learning*, pages 9786–9796. PMLR, 2020.

[48] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[49] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[50] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[51] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pages 264–273. Springer, 2024.

[52] Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. URL https://arxiv.org/abs/2403.12945.

[53] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.10125.

[54] Jai Bardhan, Patrik Drozdik, Josef Sivic, and Vladimir Petrik. Persistent robot world models: Stabilizing multi-step rollouts via reinforcement learning. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2603.25685.

[55] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[56] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.

[57] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International conference on learning representations*, 2018.

[58] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[59] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[60] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

[61] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 9902–9915. PMLR, 2022.

[62] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

[63] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

[64] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. In *arXiv preprint arXiv:2204.06125*, 2022.

[65] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, 2020.

[66] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. Infodiffusion: Representation learning using information maximizing diffusion models. *International conference on machine learning*, 2023.

[67] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*, 2021.

[68] Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Stratis Gavves. CITRIS: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, 2022.

[69] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. In *arXiv preprint arXiv:2311.15127*, 2023.

[70] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[72] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

Appendix

Supplementary material for ***Factored Representation Learning through Mechanism Diversity***

Roadmap

A **Notation** p. 16
A consolidated reference for symbols used in the theory, objectives, and experiments.

B **Formal Statements** p. 16
Formal versions of the main identifiability theorem, its temporal specialization, and the fixed-key projected-feature result.

C **Proofs** p. 18
Proofs of the formal statements in Appendix B, plus the scope of the MCL objective.

D **Experimental Details** p. 21
Disentanglement metrics, baseline descriptions, MD-VAE / MDDiff / ACG-WM architectures, training schedules, and hyperparameters.

E **PyTorch Implementation of MCL** p. 23
Primary PyTorch loss (Algorithm 1) and the full implementation including projection head and mechanism critic (Algorithm 2).

F **Additional Experimental Results** p. 24
Conditioning-set-size ablation, generation quality, latent traversals, and cross-attention maps.

G **Extended Related Work** p. 28
Why existing generative models can be factored, and a categorization of disentanglement methods by underlying principle.

H **Recommendations for Future Work** p. 29
Beyond identifiability, MCL as fine-tuning of large pretrained models, multi-objective / multi-auxiliary mechanism contrast.

A Notation

Table 4: **Notation used throughout the paper.**

Symbol	Description
Generative model and data.	
$\mathbf{x}$	Observation (data sample).
$\hat{\mathbf{x}}$	Reconstructed/generated observation produced by the model.
$\mathbf{z}$	Ground-truth latent code.
$\hat{\mathbf{z}}$	Learned latent code; identifiability is up to permutation and componentwise invertible maps.
$\mathbf{u}$	Conditioning signal: text, class label, viewpoint, regime index, or action that reshapes the generator.
$\mathbf{o}_t$	World-model observation at time t.
$\mathbf{a}_t$	World-model action at time t, playing the role of $\mathbf{u}$ in the temporal GMD setting.
Generator and identifiability.	
$g(\mathbf{z}, \mathbf{u})$ or $g_{\mathbf{u}}(\mathbf{z})$	Conditional generator family indexed by $\mathbf{u}$; GMD studies how this family varies across conditioning values.
$\hat{g}(\hat{\mathbf{z}}, \mathbf{u})$	Observationally equivalent learned generator used in the identifiability argument.
$H_{\mathbf{u}} = \hat{g}_{\mathbf{u}}^{-1} \circ g_{\mathbf{u}}$	Latent relabeling; Theorem 1 proves $H_{\mathbf{u}} = H$ for all conditions.
$Dg_{\mathbf{u}}$, J_g	Generator Jacobian of size $d_x \times d_z$; $d_x > d_z$ in Theorem 1.
DH, J_H	Jacobian of the shared latent relabeling.
$a_i(\mathbf{z}, \mathbf{u})$	Gain in $\partial_i g_{\mathbf{u}} = a_i(\mathbf{z}, \mathbf{u}) \partial_i g_0$, with $g_0 = g_{\mathbf{u}^{(0)}}$.
$\mathbf{a}_i(\mathbf{z})$	Mechanism response profile $(a_i(\mathbf{z}, \mathbf{u}))_{\mathbf{u} \in \mathcal{U}}$ for factor i.
$\Gamma(\mathbf{z})$	Reference tangent space $\operatorname{col} Dg_0(\mathbf{z})$.
d_z	Latent dimension.
d_u	Conditioning dimension.
$\lvert\mathcal{U}\rvert$	Conditioning-set size; number of distinct conditioning mechanisms available for contrast.
MCL objective.	
c_ψ	Mechanism critic whose latent gradient is contrasted against the conditioning signal.
$\mathbf{g}(\mathbf{z}, \mathbf{u})$	Mechanism gradient $\nabla_{\mathbf{z}} c_\psi(G_\theta(\mathbf{z}, \mathbf{u}), \mathbf{z}, \mathbf{u})$.
Π_g	MCL projection head mapping the mechanism gradient into the embedding space.
Π_u	MCL projection head mapping the conditioning signal into the embedding space.
$\mathbf{q}(\mathbf{z}, \mathbf{u})$	L_2-normalized projected mechanism gradient (query).
$\mathbf{k}(\mathbf{u})$	L_2-normalized projected conditioning embedding (key).
τ	InfoNCE temperature in the MCL loss.
λ_{MCL}	MCL loss weight in the full training objective.

B Formal Statements

This appendix presents the formal statements, followed by their proofs in Appendix C.

B.1 Identifiability and Its Extensions

Proposition 1 (Full Statement: Capacity Separation under an Observable-Rank Gap). *Fix two classes of conditional-law families $\mathcal{P}_{\mathrm{LRD}}$ and $\mathcal{P}_{\mathrm{GMD}}$, parameterized by open domains $\Theta_L \subset \mathbb{R}^{d_L}$ and $\Theta_G \subset \mathbb{R}^{d_G}$. Assume $\mathcal{P}_{\mathrm{LRD}} \subseteq \mathcal{P}_{\mathrm{GMD}}$. For the same finite collection of observable statistics, such as expectations of test functions at fixed conditioning values, let*

$$\Psi_L : \Theta_L \to \mathbb{R}^q, \qquad \Psi_G : \Theta_G \to \mathbb{R}^q$$

be their parameter-to-statistic maps. Suppose both maps are real analytic and, at some $\theta_G \in \Theta_G$,

$$\operatorname{rank} D\Psi_G(\theta_G) > \sup_{\theta_L \in \Theta_L} \operatorname{rank} D\Psi_L(\theta_L).$$

Then $\mathcal{P}_{\mathrm{LRD}} \subsetneq \mathcal{P}_{\mathrm{GMD}}$.

Theorem 1 (Full Statement: Identifiability under Sufficient GMD). *Let* $\mathcal{U} = \{\mathbf{u}^{(0)}, \ldots, \mathbf{u}^{(m)}\}$ *be a finite set of condition labels. Let* $\mathcal{Z} = \prod_{i=1}^{d_z} I_i$ *and* $\widehat{\mathcal{Z}} = \prod_{i=1}^{d_z} \widehat{I}_i$*, where all intervals are nonempty and open. The latent laws have positive* C^1 *densities* $p_{\mathbf{z}}$ *and* $p_{\hat{\mathbf{z}}}$ *on these domains and are independent of* $\mathbf{u}$*; their coordinates need not be independent. For every* $\mathbf{u}$*, let*

$$g_{\mathbf{u}} : \mathcal{Z} \longrightarrow \mathbb{R}^{d_x}, \qquad \hat{g}_{\mathbf{u}} : \widehat{\mathcal{Z}} \longrightarrow \mathbb{R}^{d_x}, \qquad d_x > d_z,$$

be proper C^2 *embeddings. Here proper means that inverse images of compact sets are compact. Suppose the true family satisfies Assumption 2, and the candidate family satisfies its shared-direction part:*

$$\partial_i \hat{g}_{\mathbf{u}}(\hat{\mathbf{z}}) = \hat{a}_i(\hat{\mathbf{z}}, \mathbf{u})\, \partial_i \hat{g}_0(\hat{\mathbf{z}}), \qquad \hat{a}_i(\hat{\mathbf{z}}, \mathbf{u}) \neq 0, \quad \hat{a}_i(\hat{\mathbf{z}}, \mathbf{u}^{(0)}) = 1.$$

No diversity condition on the candidate profiles is needed. Assume additionally that the reference tangent-space map

$$\Gamma : \mathcal{Z} \longrightarrow \mathrm{Gr}(d_z, d_x), \qquad \Gamma(\mathbf{z}) = \operatorname{col} Dg_0(\mathbf{z}),$$

is injective, where $\mathrm{Gr}(d_z, d_x)$ *denotes the set of* d_z*-dimensional linear subspaces of* $\mathbb{R}^{d_x}$*. If the unpaired conditional laws agree,*

$$(g_{\mathbf{u}})_{\#} p_{\mathbf{z}} = (\hat{g}_{\mathbf{u}})_{\#} p_{\hat{\mathbf{z}}} \qquad \text{for every } \mathbf{u} \in \mathcal{U},$$

then there exist one permutation π *and scalar* C^2 *diffeomorphisms* $h_i : I_{\pi(i)} \to \widehat{I}_i$ *such that*

$$\hat{g}_{\mathbf{u}}^{-1} \circ g_{\mathbf{u}} = H, \qquad H_i(\mathbf{z}) = h_i(z_{\pi(i)}), \qquad H_{\#} p_{\mathbf{z}} = p_{\hat{\mathbf{z}}}.$$

The map H *is shared across all conditions. Thus the conclusion is exact componentwise nonlinear identifiability within the stated model class, without paired observations or known composition relations between conditions.*

Proposition 2 (Full Statement: GMD with an Observed Temporal Context). *Let* $\mathcal{F}_t = \sigma(\mathbf{x}_{\leq t})$ *and fix an* $\mathcal{F}_t$*-measurable context* $U_t = e(\mathbf{x}_{\leq t})$ *taking values in the finite set* $\mathcal{U}$*, each with positive probability. Suppose*

$$\mathbf{x}_{t+1} = g_{U_t}(\mathbf{z}_{t+1}), \qquad \mathbf{z}_{t+1} \perp \mathcal{F}_t.$$

Let a candidate generator family $\hat{g}$ *with an innovation law independent of* U_t *give the same conditional laws of* $\mathbf{x}_{t+1}$ *given this same context. If the two families satisfy all assumptions of Theorem 1, then there is a single map* $H_i(\mathbf{z}) = h_i(z_{\pi(i)})$ *such that* $\hat{g}_u^{-1} \circ g_u = H$ *for every* $u \in \mathcal{U}$*. The innovation is identifiable up to a common componentwise transformation under the fixed context map.*

Scope of unpaired conditioning. The preceding results concern the separate conditional laws associated with ordinary condition labels. They do not replace these laws by a paired multi-view joint distribution or relax the embedding assumptions of Theorem 1. Appendix C.4 discusses why additional labels alone do not supply that missing information.

B.2 MCL as a Mechanism-Diversity Objective

We use the notation of Appendix A and Section 4: the host generator G_θ, the mechanism critic c_ψ, the projection heads Π_g and Π_u into a common d-dimensional embedding space, the mechanism gradient $\mathbf{g}(\mathbf{z}, \mathbf{u}) = \nabla_{\mathbf{z}} c_\psi(G_\theta(\mathbf{z}, \mathbf{u}), \mathbf{z}, \mathbf{u})$, and the L_2-normalized query and key $\mathbf{q}(\mathbf{z}, \mathbf{u}) = \Pi_g(\mathbf{g}(\mathbf{z}, \mathbf{u}))/\|\Pi_g(\mathbf{g}(\mathbf{z}, \mathbf{u}))\|$ and $\mathbf{k}(\mathbf{u}) = \Pi_u(\mathbf{u})/\|\Pi_u(\mathbf{u})\|$. For this auxiliary analysis, allow a general measurable conditioning space and consider the infinite-negative population InfoNCE objective

$$\mathcal{L}_{\mathrm{MCL}}^{\infty}(\tau) = \mathbb{E}_{(\mathbf{z},\mathbf{u}) \sim p(\mathbf{z})p(\mathbf{u})} \left[-\tfrac{\mathbf{q}(\mathbf{z},\mathbf{u})^\top \mathbf{k}(\mathbf{u})}{\tau} + \log \mathbb{E}_{\mathbf{u}' \sim p(\mathbf{u})} \exp\!\big(\tfrac{\mathbf{q}(\mathbf{z},\mathbf{u})^\top \mathbf{k}(\mathbf{u}')}{\tau}\big) \right]. \tag{2}$$

We analyze the population contrastive objective with fixed uniform keys.

Theorem 2 (Projected-feature diversity with fixed uniform keys)**.** *Fix a measurable key map* $\mathbf{k} : \mathcal{U} \to S^{d-1}$ *such that* $\mathbf{k}_{\#}p(\mathbf{u})$ *is uniform on* S^{d-1}*, with* $d \geq 2$*. Let* G_θ *be differentiable in* $\mathbf{z}$ *and* c_ψ *in* $(\mathbf{x}, \mathbf{z})$*, and assume the projected mechanism-gradient feature* $\mathbf{v}(\mathbf{z}, \mathbf{u}) = \Pi_g(\nabla_{\mathbf{z}} c_\psi(G_\theta(\mathbf{z}, \mathbf{u}), \mathbf{z}, \mathbf{u}))$ *is measurable and nonzero almost everywhere. Suppose the admissible query class contains* $\mathbf{q} = \mathbf{v}/\|\mathbf{v}\| = \mathbf{k}$ *almost everywhere. For every fixed* $\tau > 0$*, every global minimizer of* (2) *over this query class satisfies* $\mathbf{q} = \mathbf{k}$ *almost everywhere. Moreover, at each such minimizer, for every* $1 \leq r < d$*, for* $p(\mathbf{u})^{\otimes(r+1)}$*-almost every tuple* $(\mathbf{u}^{(0)}, \ldots, \mathbf{u}^{(r)})$*, the contrasts*

$$\Delta_n(\mathbf{z}) = \mathbf{v}(\mathbf{z}, \mathbf{u}^{(n)}) - \mathbf{v}(\mathbf{z}, \mathbf{u}^{(0)}), \qquad n = 1, \ldots, r, \tag{3}$$

are linearly independent for $p(\mathbf{z})$*-almost every* $\mathbf{z}$*.*

This is a statement about projected critic features under a fixed-key reference condition. Section 4 implements MCL with learned keys and finite batches, and Section 5 evaluates this training procedure.

C Proofs

C.1 Proof of Proposition 1

Proof. Write $r_L = \sup_{\theta_L} \operatorname{rank} D\Psi_L(\theta_L)$ and choose a point where $D\Psi_G$ has rank $r > r_L$. A nonzero $r \times r$ minor selects a coordinate projection $P : \mathbb{R}^q \to \mathbb{R}^r$ and r input coordinates. Holding the other input coordinates fixed, the inverse function theorem shows that $P\Psi_G(\Theta_G)$ contains a nonempty open subset of $\mathbb{R}^r$. In contrast, $\operatorname{rank} D(P\Psi_L) \leq r_L < r$ everywhere. Every point of Θ_L is therefore critical for the smooth map $P\Psi_L$, and Sard's theorem implies that $P\Psi_L(\Theta_L)$ has Lebesgue measure zero in $\mathbb{R}^r$. It cannot contain the open subset supplied by the GMD map. Hence some GMD observable statistic, and therefore some GMD conditional law, is absent from the LRD family. The assumed nesting gives strict inclusion. □

The nesting assumption must be checked for the chosen model classes. The construction $f_G(\boldsymbol{\epsilon}) = \boldsymbol{\epsilon}$, $g_G(\mathbf{u}, \mathbf{z}) = g_L(f_L(\mathbf{u}, \mathbf{z}))$ establishes inclusion when the GMD class admits the identity noise embedding and this composition with compatible dimensions. An arbitrary fixed architecture need not be closed under that construction.

C.2 Proof of Theorem 1

Proof. Write $M_{\mathbf{u}} = g_{\mathbf{u}}(\mathcal{Z})$ and $\widehat{M}_{\mathbf{u}} = \hat{g}_{\mathbf{u}}(\widehat{\mathcal{Z}})$. Proper embeddings have closed images, and positivity of the latent densities gives positive probability to every relative neighborhood of an image point. Thus each image equals the support of its conditional law. Observational equivalence implies $M_{\mathbf{u}} = \widehat{M}_{\mathbf{u}}$, so $r_{\mathbf{u}} = g_{\mathbf{u}}^{-1} \circ \hat{g}_{\mathbf{u}} : \widehat{\mathcal{Z}} \to \mathcal{Z}$ is a C^2 diffeomorphism. Only the separate conditional laws are used.

Define $D_{\mathbf{u}}(\mathbf{z}) = \operatorname{diag}(a_1(\mathbf{z}, \mathbf{u}), \ldots, a_{d_z}(\mathbf{z}, \mathbf{u}))$ and define $\widehat{D}_{\mathbf{u}}$ analogously. The shared-direction conditions give

$$Dg_{\mathbf{u}}(\mathbf{z}) = Dg_0(\mathbf{z}) D_{\mathbf{u}}(\mathbf{z}), \qquad D\hat{g}_{\mathbf{u}}(\hat{\mathbf{z}}) = D\hat{g}_0(\hat{\mathbf{z}}) \widehat{D}_{\mathbf{u}}(\hat{\mathbf{z}}).$$

Nonzero gains make the column spaces independent of $\mathbf{u}$ at a fixed latent point. Both parameterizations induce the same tangent space at $\hat{g}_{\mathbf{u}}(\hat{\mathbf{z}}) = g_{\mathbf{u}}(r_{\mathbf{u}}(\hat{\mathbf{z}}))$, hence

$$\Gamma(r_{\mathbf{u}}(\hat{\mathbf{z}})) = \operatorname{col} Dg_{\mathbf{u}}(r_{\mathbf{u}}(\hat{\mathbf{z}})) = \operatorname{col} D\hat{g}_{\mathbf{u}}(\hat{\mathbf{z}}) = \widehat{\Gamma}(\hat{\mathbf{z}}),$$

where $\widehat{\Gamma}(\hat{\mathbf{z}}) = \operatorname{col} D\hat{g}_0(\hat{\mathbf{z}})$. Injectivity of Γ gives $r_{\mathbf{u}} = r_0 =: r$ for all $\mathbf{u}$, proving the common relabeling $H = r^{-1}$. Applying g_0^{-1} to the reference law equality yields $r_{\#} p_{\hat{\mathbf{z}}} = p_{\mathbf{z}}$, equivalently $H_{\#} p_{\mathbf{z}} = p_{\hat{\mathbf{z}}}$.

We now identify the components of this common map. Put $\mathbf{z} = r(\hat{\mathbf{z}})$ and $K(\hat{\mathbf{z}}) = Dr(\hat{\mathbf{z}})$. Differentiating $\hat{g}_{\mathbf{u}} = g_{\mathbf{u}} \circ r$ at condition $\mathbf{u}$ and at the reference condition gives

$$Dg_0(\mathbf{z}) D_{\mathbf{u}}(\mathbf{z}) K(\hat{\mathbf{z}}) = D\hat{g}_{\mathbf{u}}(\hat{\mathbf{z}}) = Dg_0(\mathbf{z}) K(\hat{\mathbf{z}}) \widehat{D}_{\mathbf{u}}(\hat{\mathbf{z}}).$$

The $d_x \times d_z$ matrix $Dg_0(\mathbf{z})$ has full column rank and therefore a left inverse. Cancelling it yields the simultaneous equations

$$D_{\mathbf{u}}(\mathbf{z}) K(\hat{\mathbf{z}}) = K(\hat{\mathbf{z}}) \widehat{D}_{\mathbf{u}}(\hat{\mathbf{z}}), \qquad \mathbf{u} \in \mathcal{U}. \tag{4}$$

Entrywise, $a_i(\mathbf{z},\mathbf{u})K_{ij} = K_{ij}\hat{a}_j(\hat{\mathbf{z}},\mathbf{u})$. Multiplying the equations for rows i and k by K_{kj} and K_{ij} respectively and subtracting gives

$$\big(a_i(\mathbf{z},\mathbf{u}) - a_k(\mathbf{z},\mathbf{u})\big)K_{ij}(\hat{\mathbf{z}})K_{kj}(\hat{\mathbf{z}}) = 0 \qquad (i \neq k,\ \mathbf{u} \in \mathcal{U}). \tag{5}$$

On the dense set $\mathcal{D}$ from Assumption 2, the profiles $\mathbf{a}_i(\mathbf{z})$ and $\mathbf{a}_k(\mathbf{z})$ are linearly independent. Because their reference entries are both one, this is equivalent to the existence of a condition with $a_i(\mathbf{z},\mathbf{u}) \neq a_k(\mathbf{z},\mathbf{u})$. Equation (5) thus forces $K_{ij}K_{kj} = 0$ on $r^{-1}(\mathcal{D})$. The latter set is dense since r is a diffeomorphism; continuity extends this identity to every $\hat{\mathbf{z}}$.

Each column of K has at most one nonzero entry; invertibility gives exactly one in every row and column. The support permutation is locally constant because each nonzero entry persists on a neighborhood, excluding alternative entries in its row and column. Connectedness makes this permutation global: for one σ, $\partial_{\hat{z}_j} r_i = 0$ for $j \neq \sigma(i)$, while $\partial_{\hat{z}_{\sigma(i)}} r_i \neq 0$. Coordinate fibers are connected products of intervals, so $r_i(\hat{\mathbf{z}}) = \rho_i(\hat{z}_{\sigma(i)})$. Each ρ_i has nonvanishing derivative and is strictly monotone; surjectivity of r makes it onto I_i, hence a scalar C^2 diffeomorphism. Inverting gives $H_j(\mathbf{z}) = \rho^{-1}_{\sigma^{-1}(j)}(z_{\sigma^{-1}(j)})$. Set $\pi = \sigma^{-1}$ and $h_j = \rho^{-1}_{\sigma^{-1}(j)}$ to obtain the claimed form, shared across conditions. □

C.3 Proof of Proposition 2

Proof. Because $U_t = e(\mathbf{x}_{\leq t})$ is $\mathcal{F}_t$-measurable, innovation exogeneity $\mathbf{z}_{t+1} \perp \mathcal{F}_t$ implies $\mathbf{z}_{t+1} \perp U_t$. Each context has positive probability, so its conditional observation law is defined and agrees in the two models by assumption. The setting is therefore exactly Theorem 1, applied to the common observed label U_t. Its conclusion supplies one componentwise correspondence across all contexts. □

The context map must be shared when comparing models. If two encoders are learned independently, they may define different conditioning variables, and equality of their predictive objectives does not supply the conditional-law equivalence required above. The proposition concerns an exogenous next-step innovation, not automatic recovery of a persistent dynamical state.

C.4 Scope of Unpaired Conditioning and Multi-View Data

Ordinary condition labels identify the separate laws $p(\mathbf{x} \mid \mathbf{u}^{(n)})$. They do not identify the joint distribution $p(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ of views sharing one latent instance. In particular, an operator identity formed from $p(\mathbf{x}^{(2)}, \mathbf{x}^{(1)} \mid \mathbf{x}^{(0)})$ cannot be recovered from the separate marginals. Conditional independence of measurement noises, when assumed in a paired multi-view model, does not supply the missing pairing.

For example, let $\mathbf{z} \sim \mathcal{N}(0, I)$, let A_u be invertible, and let R_u be an orthogonal matrix for each condition. The families $g_u(\mathbf{z}) = A_u\mathbf{z}$ and $\hat{g}_u(\mathbf{z}) = A_uR_u\mathbf{z}$ have identical conditional laws, but their latent relabelings are $\hat{g}_u^{-1} \circ g_u = R_u^{-1}$. Varying R_u changes the cross-condition correspondence without changing any observed marginal, regardless of the number of labels. Theorem 1 rules out such ambiguity through its stated geometric assumptions. Relaxing those assumptions for noisy or non-injective generators requires a separate argument; paired multi-view results [35, 36] do not establish that extension in the present unpaired setting.

C.5 Proof of Theorem 2

Proof. Let σ be the uniform probability measure on S^{d-1} and put $C_{\tau,d} = \int_{S^{d-1}} \exp(\mathbf{e}^\top\mathbf{v}/\tau)\, d\sigma(\mathbf{v})$ for any unit vector $\mathbf{e}$. Rotational invariance makes this constant independent of $\mathbf{e}$. Since the key distribution is fixed and uniform, every unit query satisfies

$$\mathcal{L}^{\infty}_{\mathrm{MCL}} = -\frac{1}{\tau}\mathbb{E}[\mathbf{q}^\top\mathbf{k}] + \log C_{\tau,d} = -\frac{1}{\tau} + \log C_{\tau,d} + \frac{1}{2\tau}\mathbb{E}\|\mathbf{q}-\mathbf{k}\|^2.$$

Realizability attains the lower bound, and equality holds exactly when $\mathbf{q} = \mathbf{k}$ almost everywhere.

Independent uniform sphere vectors are linearly independent almost surely whenever their number is at most d: conditional on any independent preceding vectors, their span is a proper linear subspace

whose intersection with the sphere has zero surface measure. Thus, for almost every tuple of $r+1 \leq d$ conditions, the keys $\mathbf{k}_n = \mathbf{k}(\mathbf{u}^{(n)})$ are linearly independent. Fubini's theorem also ensures that, for almost every such tuple, alignment and nonvanishing hold at all its entries for $p(\mathbf{z})$-almost every $\mathbf{z}$. At each such $\mathbf{z}$, write $\mathbf{v}(\mathbf{z}, \mathbf{u}^{(n)}) = \alpha_n(\mathbf{z})\mathbf{k}_n$ with $\alpha_n > 0$. If $\sum_{n=1}^{r} \beta_n \Delta_n = 0$, then

$$\sum_{n=1}^{r} \beta_n \alpha_n \mathbf{k}_n - \alpha_0 \Big(\sum_{n=1}^{r} \beta_n \Big) \mathbf{k}_0 = 0.$$

Key independence gives $\beta_n \alpha_n = 0$ for every n, hence all $\beta_n = 0$. □

C.6 Scope of the Mechanism-Contrastive Objective

Theorem 2 concerns diversity of projected critic features. Theorem 1 instead requires shared factor response directions, distinct generator gain profiles, and an injective reference tangent-space map. These are different mathematical properties, and the contrastive result alone does not imply the identifiability assumptions.

The distinction follows directly from the chain rule. Evaluating derivatives of c_ψ at $(G_\theta(\mathbf{z}, \mathbf{u}), \mathbf{z}, \mathbf{u})$ gives

$$\mathbf{g}(\mathbf{z}, \mathbf{u}) = D_{\mathbf{z}} G_\theta(\mathbf{z}, \mathbf{u})^\top \nabla_{\mathbf{x}} c_\psi + \partial_{\mathbf{z}} c_\psi. \tag{6}$$

The second term is the critic's explicit latent dependence. For example, $c_\psi(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \mathbf{z}^\top b(\mathbf{u})$ yields $\mathbf{g} = b(\mathbf{u})$ independently of the generator. Hence condition-discriminative critic features can arise without changing $D_{\mathbf{z}} G_\theta$. A reconstruction or denoising objective for the host generator does not, by itself, impose score faithfulness on this auxiliary critic.

MCL is therefore used as a practical mechanism-contrastive training signal, with its generator-level effects assessed empirically. Generator-only diagnostics must exclude the critic and projection heads and specify their differentiated inputs and aggregation. Conditioning sensitivities $D_{\mathbf{u}} G_\theta$ measure responsiveness to $\mathbf{u}$, whereas Theorem 1 concerns the geometry of $D_{\mathbf{z}} G_\theta$ and the reference tangent-space map. Establishing an architectural or objective-level implication from practical MCL to all assumptions of Theorem 1 remains open.

C.7 Comparison with Contrastive Objectives

Contrastive losses share the InfoNCE template but differ in the features they compare and the guarantees available under their assumptions. Table 5 compares MCL against four representative families. MCL contrasts critic-derived mechanism gradients $\nabla_{\mathbf{z}} c_\psi(G(\mathbf{z}, \mathbf{u}), \mathbf{z}, \mathbf{u})$ against the conditioning signal, with negatives obtained by shuffling $\mathbf{u}$ within a batch. Theorem 2 establishes projected-feature diversity under fixed uniform keys.

Table 5: **Contrastive objective families.** Columns list the positive pair, the source of negatives, the object contrasted, and what the optimum targets, for five representative families.

Family	Positive pair	Negatives	Object contrasted	Optimum target
Augmentation [SimCLR/MoCo; 55]	$\big(f(\mathbf{x}), f(\widetilde{\mathbf{x}})\big)$, $\widetilde{\mathbf{x}}$ an augmentation of $\mathbf{x}$	In-batch other images	Image representation $f(\mathbf{x})$	Augmentation-invariant similarity
Predictive [CPC; 16]	$\big(c_t, f(\mathbf{x}_{t+k})\big)$, context vs. future	In-batch other futures	Context vs. future representation	MI lower bound $I(c_t; \mathbf{x}_{t+k})$
Time-contrastive [TCL; 10]	($\mathbf{x}$, segment label)	Other segment labels	Encoder of $\mathbf{x}$	Nonlinear ICA / LRD identifiability
Density-ratio / NCE [56]	Data $\mathbf{x}$ vs. noise $\widetilde{\mathbf{x}}$	Noise samples	Log-density-ratio $\log p_{\text{data}}/p_{\text{noise}}$	Density / score estimation
Mechanistic MCL (ours)	$\big(\mathbf{g}(\mathbf{z}, \mathbf{u}), \mathbf{u}\big)$	Shuffled $\mathbf{u}_j$ in batch	Mechanism gradient vs. conditioning	Projected-feature diversity with fixed keys, Theorem 2

D Experimental Details

D.1 Disentanglement Metrics

FactorVAE score [22]. For each vote, fix one ground-truth factor k and vary the others. Encode the resulting batch and compute each active latent coordinate's variance normalized by its variance over the full data distribution. The coordinate with the smallest normalized variance predicts the fixed factor through a majority-vote classifier. The reported score is the classifier's accuracy on held-out votes.

DCI disentanglement [57]. Train a predictor from $\mathbf{z}$ to each factor k and extract the feature importance matrix $R \in \mathbb{R}^{d \times K}$. The disentanglement score is

$$D = 1 - \frac{1}{d} \sum_{j=1}^{d} H\left(\frac{R_{j\cdot}}{\sum_k R_{jk}} \right) \Big/ \log K, \tag{7}$$

where $H(\cdot)$ is the entropy. $D = 1$ when each latent dimension encodes exactly one factor.

D.2 Baselines for Concept Disentanglement

We expand the baseline descriptions for Section 5.2 (Table 1).

VAE-based. **FactorVAE** [22] adds a total-correlation penalty on the aggregated posterior to encourage statistically independent latents. **β-TCVAE** [23] decomposes the KL term and upweights the total-correlation component, isolating the disentanglement-relevant pressure of β-VAE.

GAN-based. **InfoGAN-CR** [45] extends InfoGAN with a contrastive regulariser on the inferred factors. **GANSpace** [46] discovers disentangled directions *post-hoc* via PCA on early-layer GAN activations. **LatentDisco** [47] learns directions in a pretrained GAN's latent space using a self-supervised classifier. **DisCo** [43] is a contrastive method on a pretrained generator: it samples paired latent walks and trains an encoder to detect the active factor.

Diffusion-based. **DisDiff** [44] associates each latent with a per-factor sub-gradient field of the score network. **EncDiff** [40] routes each scalar latent through an independent MLP and injects the resulting tokens via cross-attention into the U-Net. **MDDiff (ours)** uses this tokenized diffusion scaffold, adds $\mathbf{u}$-conditioned decoding to form the MDDiff pretraining model, and then fine-tunes that checkpoint with MCL to encourage condition-discriminative mechanism-gradient features.

D.3 MD-VAE (Simulations)

Synthetic data-generating process. We follow the standard nonlinear ICA simulation protocol [7, 8] and instantiate it inside the GMD regime. For each run we draw a latent dimension $d_z \in \{5, 10, 20\}$ and a conditioning-set size $|\mathcal{U}| \in \{2, 4, 8, 16, 32\}$ (default $|\mathcal{U}| = 8$). Each simulated dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{u}_i)\}_{i=1}^{N}$ with $N = 50{,}000$ is generated as follows. The *conditioning signal* $\mathbf{u}_i$ is drawn uniformly from a fixed alphabet $\mathcal{U} = \{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(|\mathcal{U}|)}\}$, where each $\mathbf{u}^{(n)} \in \mathbb{R}^{d_u}$ ($d_u = 16$) is sampled once at the start of the run from $\mathcal{N}(0, \mathbf{I}_{d_u})$. The *latent* $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}_{d_z})$ is sampled independently of $\mathbf{u}_i$, instantiating the GMD assumption $\mathbf{z} \perp \mathbf{u}$; this is the key difference from the iVAE [8] simulation, in which $p(\mathbf{z} \mid \mathbf{u})$ shifts. The *mechanism-diverse generator* is $g(\mathbf{z}, \mathbf{u}) = h \circ \big(\mathbf{z} \odot \mathbf{m}(\mathbf{u}) + \mathbf{b}(\mathbf{u})\big)$, where $\mathbf{m}, \mathbf{b} : \mathbb{R}^{d_u} \to \mathbb{R}^{d_z}$ are two-layer MLPs (width 64, leaky-ReLU) initialized so that $\mathbf{m}(\mathbf{u}^{(n)})$ varies meaningfully across n, and h is a fixed invertible 3-layer leaky-ReLU MLP (width $\max(64, 4d_z)$); the componentwise product $\mathbf{z} \odot \mathbf{m}(\mathbf{u})$ modulates the inputs to the nonlinear mixing map. This construction provides controlled mechanism modulation for the simulation. Finally, observations are $\mathbf{x}_i = g(\mathbf{z}_i, \mathbf{u}_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, 0.01\,\mathbf{I})$. We use an $80\%/20\%$ train/test split, repeating each configuration over 30 random seeds.

Models. All three baselines share the same encoder–decoder backbone (3-layer leaky-ReLU MLPs, hidden width $\max(64, 4d_z)$, latent dim d_z) and differ only in how $\mathbf{u}$ is incorporated. **β-VAE** [21] is a standard VAE with KL weight $\beta = 4$ in which $\mathbf{u}$ is fed to neither encoder nor

decoder. **iVAE** [8] uses a $\mathbf{u}$-conditional encoder $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{u})$ and a $\mathbf{u}$-conditional Gaussian prior $p(\mathbf{z} \mid \mathbf{u}) = \mathcal{N}(\mu_\lambda(\mathbf{u}), \mathrm{diag}(\sigma^2_\lambda(\mathbf{u})))$ implemented by a 2-layer MLP, with a decoder that is unconditioned in $\mathbf{u}$ (this is exactly the LRD regime). **MD-VAE** (ours) keeps the prior $\mathbf{u}$-independent, $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$, and feeds $\mathbf{u}$ to both the encoder $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{u})$ and the decoder $p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{u})$, training with the conditional ELBO of Equation (1). We optimize with Adam (learning rate 10^{-3}, batch size 256) for 200 epochs.

Evaluation. After training we recover latents $\hat{\mathbf{z}}_i = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i,\mathbf{u}_i)}[\mathbf{z}]$ on the test split and compute MCC by solving the linear assignment problem between $\hat{\mathbf{z}}$ and the true $\mathbf{z}$ in absolute Pearson correlation, following Khemakhem et al. [8], Hyvarinen and Morioka [10].

D.4 MDDiff (Image Disentanglement)

Architecture. MDDiff builds on EncDiff [40] with two structural additions that together make it a bona fide GMD model. The shared backbone is unchanged from EncDiff: an `Encoder4` CNN (hidden width 128, four stride-2 convolutional layers) maps a 64×64 RGB image to $K=20$ scalar concept factors; each scalar is expanded by an independent three-layer MLP ($1 \to 64 \to 128 \to 16$, ELU activations) into a 16-dim concept token, so each token corresponds to exactly one semantic factor. The denoising backbone is a U-Net with base channel width 64, channel multipliers $[1, 2, 4, 4]$, 2 residual blocks per resolution, and 8-head spatial cross-attention injecting the 20 concept tokens at all three downsampled resolutions.

On top of this backbone we make two additions required for the GMD interpretation. First, we condition the *VAE decoder* on the concept variable $\mathbf{u}$: the original EncDiff feeds the concept tokens only through the U-Net cross-attention, so the VAE decoder that maps the diffusion latent back to pixels is unconditional in $\mathbf{u}$ and the concept variable never enters the generator $g(\mathbf{z}, \mathbf{u})$ in the strict sense of Assumption 2. We therefore inject $\mathbf{u}$ as an auxiliary input to the VAE decoder so that it becomes $g(\mathbf{z}, \mathbf{u})$ with $\mathbf{z}$ the diffusion latent and $\mathbf{u}$ the concept token sequence; concretely, we concatenate the per-token features to the spatial feature map at each upsampling stage of the decoder, mirroring the cross-attention pattern in the U-Net. Second, we add a *mechanism critic* $c_\psi(\hat{\mathbf{x}}, \mathbf{z}, \mathbf{u})$ scoring reconstruction–latent–concept tuples, together with two MLP projection heads Π_g, Π_u ($\to$ 128-dim L_2-normalized embeddings) that map the critic gradient $\nabla_{\mathbf{z}} c_\psi$ and the concept variable into a shared embedding space, supporting the InfoNCE objective. The two additions together add roughly 265K parameters on top of the shared backbone.

MCL Objective. We compared five MCL variants: NCE, InfoNCE, Fisher score matching, denoising score matching, and Jacobian-based InfoNCE. InfoNCE consistently achieved the highest DCI disentanglement and FactorVAE scores across all three datasets. The other objectives either suffer from training instability (Fisher, Jacobian) or weaker gradient signal when the critic operates on high-dimensional image reconstructions (NCE, denoising SM). We therefore adopt InfoNCE as our final objective:

$$\mathcal{L}_{\mathrm{MCL}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\big(\Pi_g(\mathbf{g}_i)^\top \Pi_u(\mathbf{u}_i)/\tau\big)}{\sum_{j=1}^{B} \exp\big(\Pi_g(\mathbf{g}_i)^\top \Pi_u(\mathbf{u}_j)/\tau\big)}, \tag{8}$$

where $\mathbf{g}_i = \nabla_{\mathbf{z}_i}\, c_\psi(\hat{\mathbf{x}}_i, \mathbf{z}_i, \mathbf{u}_i)$ is the critic gradient with respect to the latent, τ=0.1 is the temperature, and negatives are drawn from other samples in the same mini-batch via latent shuffling. The total training loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{diffusion}} + \lambda_{\mathrm{recon}}\, \mathcal{L}_{\mathrm{recon}} + \lambda_{\mathrm{MCL}}\, \mathcal{L}_{\mathrm{MCL}}, \tag{9}$$

where $\mathcal{L}_{\mathrm{recon}}$ is the pixel-level VQ-VAE reconstruction loss weighted by λ_{recon}=0.1.

Training. Training proceeds in two stages on a single NVIDIA H100 GPU. *Stage 1* pretrains MDDiff as a conditional-generation model with the diffusion loss only for 10 epochs (AdamW, base learning rate 2×10^{-6}, 10,000-step linear warmup). *Stage 2* fine-tunes the pretrained MDDiff checkpoint with MCL, optimizing all components jointly under the full objective $\mathcal{L}$ for an additional 10 epochs (base learning rate 2×10^{-7}, 1,000-step warmup). Both stages use batch size 128 and T=1000 diffusion timesteps with a linear noise schedule ($\beta_{\min}$=0.0015, $\beta_{\max}$=0.0155). Stage 2 fine-tuning takes approximately 3 hours on Cars3D, 4 hours on Shapes3D, and 7 hours on MPI3D. We set λ_{MCL}=10^{-3} for Shapes3D and Cars3D, and λ_{MCL}=10^{-4} for MPI3D.

D.5 Action-Conditioned Generative World Model

Pipeline. Wan-ACG and Wan-LACG are built on a large pretrained video generative model [50], equipped with four components that together realize the forward action-conditioned generative model $\mathbf{x}_{t+1} = g_{\text{fwd}}(\mathbf{z}_t, \mathbf{u}_t)$ on which MCL operates: (i) a frozen causal VAE encoder E_{vae} that maps a T-frame video window to a latent video sequence $v_{1:F} = E_{\text{vae}}(\mathbf{x}_{1:T})$; (ii) an inverse dynamics model (IDM) I_ϕ that produces, for each frame transition, a diagonal-Gaussian posterior $q_\phi(z_t \mid \mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N}(\mu_t, \text{diag}(\sigma_t^2))$ over a d_z-dimensional latent action; (iii) an action-interface module A_ψ that lifts the frame-rate latent action sequence $z_{1:T}$ to FDM-aligned conditioning signals $c_{1:F} = A_\psi(z_{1:T})$, injected into the FDM through AdaLN modulation; and (iv) a forward dynamics model (FDM) g_θ, initialized from the pretrained video-planner backbone, that performs flow matching $\hat{u}_\tau = g_\theta(\tilde{v}_\tau, \tau, z_{1:T})$ to predict future latent video dynamics conditioned on the latent action.

Mapping to the GMD setup. In our notation, the conditioning signal is the latent action, $\mathbf{u}_t := z_t$ (or, when an external command interface is required, a controller-predicted latent action $\hat{z}_t$), and the observation is the next latent video frame, $\mathbf{x}_{t+1} := v_{f+1}$. The action-conditioned FDM is the GMD generator $g(\mathbf{z}_t, \mathbf{u}_t)$: action-driven conditioning enters through AdaLN; the diffusion latent enters through $\tilde{v}_\tau$.

Our addition: MCL on the action conditioning. We attach the MCL objective from Section 4 to the action-conditioning channel of the FDM. A mechanism critic $c_\psi(\hat{\mathbf{x}}_{t+1}, z_t, \mathbf{u}_t)$ scores how strongly the realized forward mechanism depends on z_t, the mechanism gradient $\mathbf{g}_t = \nabla_{z_t} c_\psi(\hat{\mathbf{x}}_{t+1}, z_t, \mathbf{u}_t)$ is contrasted via InfoNCE against the projected latent-action embedding $\Pi_u(\mathbf{u}_t)$, and the resulting $\mathcal{L}_{\text{MCL}}$ is added to the host objective with weight λ_{MCL}:

$$\mathcal{L}_{\text{total}} \;=\; L_{\text{rec}} + \beta L_{\text{KL}} + \lambda_{\text{adv}} L_{\text{GRL}} + \lambda_{\text{MCL}}\, \mathcal{L}_{\text{MCL}},$$

where L_{rec} is the flow-matching reconstruction loss, L_{KL} is a KL regulariser on the IDM posterior, and L_{GRL} is a gradient-reversal embodiment-invariance term used to encourage the latent action to ignore embodiment-specific nuisance. The IDM, action interface, FDM, and MCL projection heads are trained jointly; the causal VAE remains frozen.

What MCL adds, and what we test. MCL adds a condition-discriminative objective on critic-derived gradients to the host prediction loss. The paired architecture comparisons in Section 5.3 test its effect on visual prediction quality.

E PyTorch Implementation of MCL

For reproducibility we provide a self-contained PyTorch implementation of the Mechanistic Contrastive Learning loss. We first present the *primary* loss (Algorithm 1) that captures the core mechanism-gradient → projection → InfoNCE pipeline used throughout the paper, then give the *full* implementation (Algorithm 2) including the projection-head module `MLPProj` and the mechanism-critic module `MechanismCritic` that we use in the image and world-model experiments. The same loss applies to every host model in our experiments (MD-VAE, MDDiff, ACG-WM); only the host components change. The loss adds to the host's reconstruction objective with weight λ_{MCL} as $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{MCL}}\, \mathcal{L}_{\text{MCL}}$.

E.1 Primary MCL Loss

E.2 Full Code

The full implementation we use in practice adds the two-layer projection head `MLPProj`, the default mechanism critic `MechanismCritic`, and the InfoNCE wrapper used in our image and world-model experiments.

Algorithm 1. Primary PyTorch implementation of the MCL loss. The host generator `decoder_G` produces $\hat{\mathbf{x}}$ from a batch of latents and conditioning signals; the critic scores how strongly the realized mechanism depends on $\mathbf{z}$; the autograd of the critic w.r.t. $\mathbf{z}$ yields the per-sample mechanism gradient, which is contrasted via InfoNCE against the projected conditioning embedding.

```
def mcl_loss(decoder_G, z, u, *,
             critic, Pi_g, Pi_u, tau=0.1):
    """
    decoder_G : callable (z, u) -> x_hat
    z         : latent batch, shape (B, ...)
    u         : conditioning batch, shape (B, d_u)
    critic    : (x_hat, z, u) -> mechanism score per sample
    Pi_g, Pi_u: projection heads to shared embedding space
    """
    z = z.requires_grad_(True)
    x_hat = decoder_G(z, u)
    s = critic(x_hat, z, u)

    # mechanism gradient w.r.t. the latent
    g = torch.autograd.grad(s.sum(), z, create_graph=True)[0]

    # L2-normalized projections
    q = Pi_g(g.flatten(1));  q = q / q.norm(dim=1, keepdim=
        True)
    k = Pi_u(u);             k = k / k.norm(dim=1, keepdim=
        True)

    # B x B InfoNCE: positives on the diagonal
    logits = q @ k.t() / tau
    labels = torch.arange(q.size(0), device=q.device)
    return F.cross_entropy(logits, labels)
```

F Additional Experimental Results

F.1 Effect of Mechanism Diversity ($|\mathcal{U}|$ ablation)

We examine empirically how conditioning-set size affects factor recovery. Table 6 reports MD-VAE's median MCC across $|\mathcal{U}| \in \{2, 4, 8, 16, 32\}$ at $d_z \in \{5, 10, 20\}$, using the settings of Appendix D.3.

Table 6: **Effect of conditioning-set size $|\mathcal{U}|$ on MD-VAE's median MCC (↑, 30 seeds).**

	$\lvert\mathcal{U}\rvert = 2$	$\lvert\mathcal{U}\rvert = 4$	$\lvert\mathcal{U}\rvert = 8$	$\lvert\mathcal{U}\rvert = 16$	$\lvert\mathcal{U}\rvert = 32$
$d_z = 5$	0.51	0.79	0.96	0.97	0.97
$d_z = 10$	0.43	0.62	0.87	0.93	0.94
$d_z = 20$	0.39	0.51	0.81	0.86	0.92

MCC increases with $|\mathcal{U}|$ in these experiments, and the higher-dimensional settings continue to improve over a wider range of condition counts.

F.2 Generation Quality of MDDiff

A representation is only useful if the generator that produces it remains a competent image model. Table 7 reports FID (sample quality) and LPIPS (reconstruction fidelity) for MDDiff and the strongest diffusion baseline EncDiff [40] on the three benchmarks of Section 5.2. MDDiff preserves generation quality while delivering the disentanglement gains reported in Table 1; on `Cars3D` it actually *improves* FID and LPIPS, and on `Shapes3D` the reported values are close. `MPI3D` shows a modest FID gap while LPIPS is unchanged.

Algorithm 2. Full PyTorch implementation of MCL: the projection-head module, default mechanism-critic module, and the InfoNCE mechanism-gradient loss with optional separate keys.

```
import torch
import torch.nn as nn
import torch.nn.functional as F

def l2norm(x: torch.Tensor, eps: float = 1e-8) -> torch.Tensor:
    return x / (x.norm(dim=1, keepdim=True) + eps)

def info_nce_from_qk(q, k, tau: float = 0.1) -> torch.Tensor:
    logits = (q @ k.t()) / (tau + 1e-12)
    labels = torch.arange(q.size(0), device=q.device)
    return F.cross_entropy(logits, labels)

class MLPProj(nn.Module):
    def __init__(self, in_dim, out_dim=128, layernorm=False):
        super().__init__()
        layers = []
        if layernorm:
            layers.append(nn.LayerNorm(in_dim))
        layers += [nn.Linear(in_dim, out_dim),
                   nn.ReLU(inplace=True),
                   nn.Linear(out_dim, out_dim)]
        self.net = nn.Sequential(*layers)

    def forward(self, x):
        return self.net(x)

class MechanismCritic(nn.Module):
    def __init__(self, z_shape=(3, 16, 16), u_dim=20, hidden=256):
        super().__init__()
        zc, zh, zw = z_shape
        self.img = nn.Sequential(
            nn.Conv2d(3, 64, 4, 2, 1), nn.ReLU(True),
            nn.Conv2d(64, 128, 4, 2, 1), nn.ReLU(True),
            nn.AdaptiveAvgPool2d(1),
        )
        self.z_fc = nn.Linear(zc * zh * zw, hidden)
        self.u_fc = nn.Linear(u_dim, hidden)
        self.out  = nn.Sequential(nn.ReLU(True), nn.Linear(hidden, 1))

    def forward(self, x_hat, z, u):
        img_feat = self.img(x_hat).flatten(1)
        z_feat = self.z_fc(z.flatten(1))
        u_feat = self.u_fc(u)
        if img_feat.size(1) < z_feat.size(1):
            img_feat = F.pad(img_feat,
                             (0, z_feat.size(1) - img_feat.size(1)))
        else:
            img_feat = img_feat[:, :z_feat.size(1)]
        return self.out(z_feat + u_feat + img_feat).squeeze(1)

def mcl_infonce_mechgrad_loss(decoder_G, z, u_key, *,
                              u_for_G=None,
                              critic, Pi_g, Pi_u,
                              tau: float = 0.1,
                              create_graph: bool = False):
    """InfoNCE mechanism-gradient loss with optional separate keys.

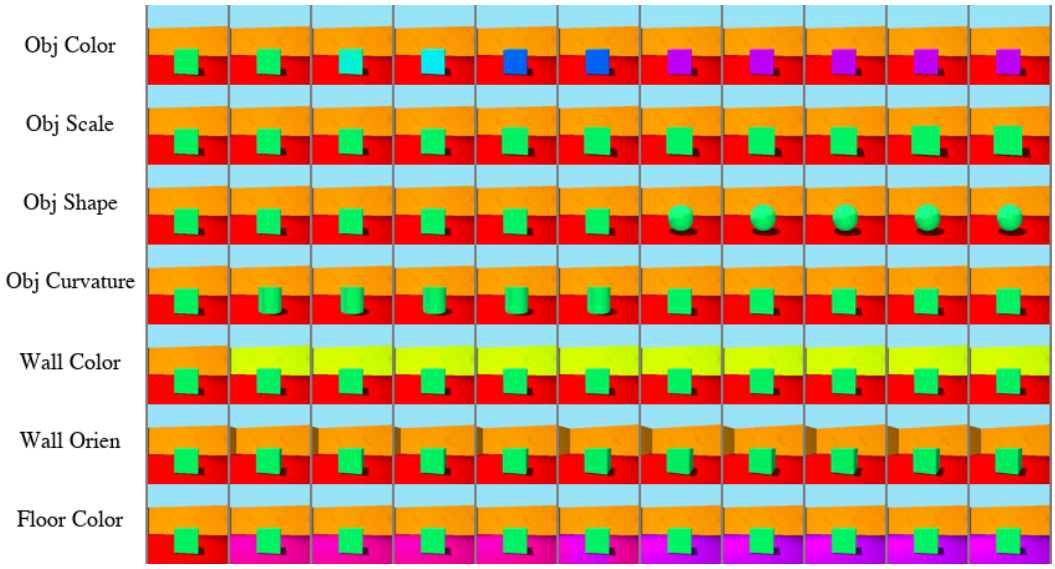
    u_key   : auxiliary used as the contrastive key
    u_for_G : conditioning fed to decoder/critic (defaults to u_key)
    """
    u = u_key
    uG = u_for_G if u_for_G is not None else u
    z = z.requires_grad_(True)
    x_hat = decoder_G(z, uG)
    s = critic(x_hat, z, uG)
    g = torch.autograd.grad(s.sum(), z, create_graph=create_graph)[0]

    q = l2norm(Pi_g(g.flatten(1)))
    k = l2norm(Pi_u(u))
    return info_nce_from_qk(q, k, tau=tau)
```

Table 7: **Generation quality (FID ↓, LPIPS ↓).**

	Cars3D		Shapes3D		MPI3D	
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
EncDiff	62.31± 3.28	0.07± 0.03	5.33± 0.68	0.01± 0.01	4.06± 0.60	0.01± 0.01
MDDiff (Ours)	**39.32**± 6.67	**0.03**± 0.02	5.52± 0.78	0.01± 0.01	7.10± 2.12	0.01± 0.01

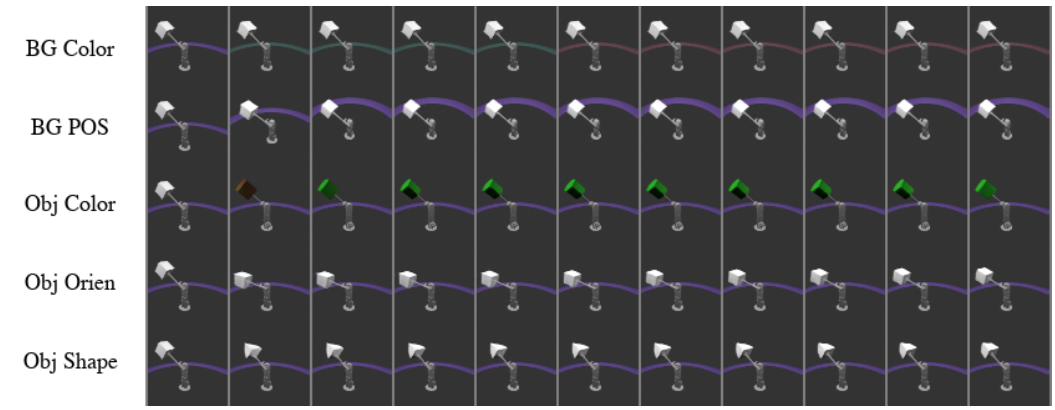
F.3 Latent Traversals



(a) Shapes3D



(b) Cars3D



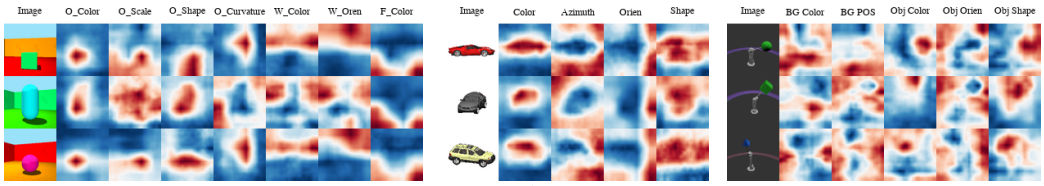
(c) MPI3D

Figure 8: **Latent traversals on all three benchmarks.** Each row varies a single MDDiff latent dimension while keeping the others fixed; columns sweep the dimension across its range.

Latent traversals offer a complementary view of factoredness to the factor-swapping results in the main paper (Figure 6): holding all other dimensions fixed and sweeping one dimension across its range should produce smooth, single-factor change in the generated image. Figure 8 reports MDDiff traversals on all three benchmarks; each row corresponds to one latent dimension and isolates one semantic factor (wall colour, floor colour, object colour, shape, orientation, scale on `Shapes3D`; analogous factors on `Cars3D` and `MPI3D`).

F.4 Cross-attention Maps

Cross-attention maps complement latent traversals by exposing how mechanism diversity manifests *inside* the generator: each concept token (one per latent dimension) modulates the denoising U-Net through cross-attention, and each token's attention should localise to the spatial region of its factor. Figure 9 visualises the cross-attention weights between concept tokens (columns) and spatial locations across the three benchmarks; each token attends to the spatial region corresponding to its factor, evidence that MCL produces semantically meaningful, spatially-localised modulation in the generator.



(a) `Shapes3D` (b) `Cars3D` (c) `MPI3D`

Figure 9: **Cross-attention maps on all three benchmarks.** Columns show concept tokens; rows show spatial locations in the denoising network.

G Extended Related Work

G.1 Relation to Existing Conditional Generators

Theorem 1 provides sufficient geometric conditions under which conditional laws identify latent components. Conditional image models [1, 30, 40, 44, 58] and world models [33, 37, 59, 60] motivate the broader setting because prompts, concept tokens, and actions modulate their generators. Decision-as-generation methods likewise condition trajectories on returns or skills [61–63]. These models motivate the conditional-generation setting; Theorem 1 characterizes latent recovery under its stated geometric assumptions.

G.2 Theoretical Foundations of LRD: Auxiliary-Variable Identifiability

Hyvarinen et al. [7] study a fixed smooth invertible mixing $\mathbf{x} = g(\mathbf{z})$ with conditionally independent sources, $p(\mathbf{z} \mid \mathbf{u}) = \prod_i p_i(z_i \mid \mathbf{u})$. Their Theorem 1 obtains componentwise nonlinear recovery under sufficient variation of the first and second derivatives of the conditional log-densities, an invertible smooth feature map, and the stated population contrastive-estimation conditions. Assumption 1 summarizes its variability requirement. Their separate exponential-family result, Theorem 3, identifies sufficient-statistic features only up to a general linear transformation; the sufficient statistics need not be invertible functions of individual sources. Our Theorem 1 instead keeps the latent law fixed and locates diversity in generator response profiles, together with explicit geometric restrictions.

G.3 Theoretical Foundations of Regularization: Sparsity-Based Identifiability

Zheng et al. [25] obtain nonlinear-ICA identifiability without auxiliary variables by combining marginally independent sources with restrictions on the mixing process. Their Theorem 1 uses smooth invertible mixing, a variability condition on Jacobian rows, a sparsity comparison between the estimated and true mixing functions, and a structural condition that isolates each source through intersections of observation-parent sets. Under these assumptions the latent correspondence is a permutation followed by componentwise invertible transformations. This result is stronger than a

requirement that the Jacobian merely have few nonzero entries or distinguishable columns; a generic L_1 or L_2 penalty does not by itself establish all its assumptions. The contrast with our result concerns the source of structure: this approach constrains Jacobian supports, whereas Assumption 2 constrains response directions and their variation across conditions.

G.4 Categorizing Methods by Principle

Table 8 organizes representative methods by their use of auxiliary-dependent latent distributions, structural regularization, and conditional generation. The categories distinguish where auxiliary variables enter the model and which properties its objective encourages. MCL explicitly contrasts critic-derived mechanism features across conditions; Appendix C.6 explains why this objective alone does not establish Assumption 2.

LRD ∩ GMD. Methods in this intersection let $\mathbf{u}$ enter both the latent distribution and the generator. Conditional VAEs [4] have this structure by design: the prior $p(\mathbf{z} \mid \mathbf{u})$ and the decoder $p(\mathbf{x} \mid \mathbf{z}, \mathbf{u})$ both depend on $\mathbf{u}$. Classifier-guided diffusion [31] adds a classifier gradient to the sampling trajectory, shifting the effective latent distribution, on top of a class-conditional denoiser that reshapes the generator. DALL-E 2 [64] factorizes generation as $P(\mathbf{x} \mid \mathbf{y}) = P(\mathbf{x} \mid \mathbf{z}, \mathbf{y})\, P(\mathbf{z} \mid \mathbf{y})$: the text caption $\mathbf{y}$ enters both the diffusion prior over CLIP embeddings and the decoder. VQ-VAE-2 [39] combines all three principles: vector quantization, top-level codes conditioning the autoregressive prior over bottom-level codes, and top-level codes entering the feed-forward decoder.

Regularization ∩ GMD. Methods here apply explicit or implicit regularization to a conditional generator. EncDiff [40] conditions a latent diffusion model on concept tokens via cross-attention, with the diffusion information bottleneck serving as implicit regularization. DisDiff [44] decomposes the score into per-factor sub-gradient fields, a structural sparsity constraint, and conditions each field on a discovered factor. The Hessian Penalty [65] regularizes the off-diagonal Hessian of a GAN generator with respect to its latent input, encouraging axis-aligned disentanglement. VQ-VAE [38] combines vector quantization with a decoder conditioned on discrete codes. InfoDiffusion [66] adds a mutual-information maximization penalty to a latent-conditioned diffusion model to prevent posterior collapse.

LRD ∩ Regularization. Methods here combine auxiliary-variable-driven distributional shifts with structural constraints. Lachapelle et al. [24] use temporal or interventional auxiliary variables together with binary-mask sparsity on the causal graph; neither alone suffices for identifiability. SlowVAE [67] uses temporal context as the auxiliary variable and a Laplace sparse prior on temporal differences. CITRIS [68] observes intervention targets and constrains the transition model to respect a factored causal structure. TDRL [12] leverages distribution shifts across environments with a modular, sparse mechanism-shift assumption.

H Recommendations for Future Work

This work studies ***mechanism diversity*** as part of a sufficient geometric criterion for factored generative models, and introduces MCL as a related training objective. The following directions concern extending the theorem's scope and understanding the practical objective's effects.

H.1 Beyond Factored Generative Models

Motivation. The notion of *factoredness* in Section 3.2, inherited from nonlinear ICA, is a stricter target than what most modern conditional generators are built or evaluated against. Large text-to-image diffusion [1, 30, 58], controllable video diffusion [69], visual world models [33, 60], and multi-modal generators routinely operate without ground-truth factors yet still benefit from compositional, controllable behaviour at the conditioning interface.

Conjecture. Condition-discriminative mechanism-gradient objectives may improve compositional generalization, prompt following, and reuse of conditioning slots even when a componentwise identifiability guarantee is unavailable. Controlled comparisons and generator-only measurements can test this conjecture.

Table 8: **Categorizing methods by principle.** • = *active* (an explicit loss term encourages the principle); ○ = *passive* (satisfied by model architecture/design only, no dedicated loss).

Region	Method	LRD	Reg.	GMD	Mechanism
LRD only	iVAE [8]	○			Conditional prior $p(\mathbf{z}\|\mathbf{u})$ (architectural)
	TCL [10]	•			Contrastive classification loss over time segments
	CPC [16]	•			InfoNCE contrastive loss over temporal windows
Reg. only	β-VAE [21]		•		Upweighted KL penalty ($\beta > 1$)
	FactorVAE [22]		•		TC penalty via learned discriminator
	Sparse ICA [25]		•		L_1 sparsity on generator Jacobian
	Tripod [27]		•		Quantization + MI + Hessian penalty losses
Active GMD	MDDiff / MCL (Ours)			•	Contrastive loss on critic-derived mechanism gradients
	MD-VAE (Ours)			•	Conditional VAE with mechanism-diverse decoder
LRD ∩ GMD	CVAE [4]	○		○	Prior $p(\mathbf{z}\|\mathbf{u})$ + decoder $p(\mathbf{x}\|\mathbf{z},\mathbf{u})$ (both architectural)
	Classifier Guidance [31]	•		○	Classifier loss shifts sampling; denoiser cond. on y
	DALL-E 2 [64]	○		○	Diffusion prior $P(\mathbf{z}\|\mathbf{y})$ + decoder $P(\mathbf{x}\|\mathbf{z},\mathbf{y})$
Reg. ∩ GMD	EncDiff [40]		○	○	Architectural bottleneck + cross-attn conditioning
	DisDiff [44]		•	○	Disentangling loss + per-factor sub-gradient fields
	Hessian Penalty [65]		•	○	Hessian reg. loss + class-conditional GAN
	VQ-VAE [38]		•	○	VQ + commitment losses + code-conditioned decoder
	InfoDiffusion [66]		•	○	MI maximization loss + latent-conditioned diffusion
All three	VQ-VAE-2 [39]	○	•	○	VQ loss (active reg.) + cond. prior + cond. decoder
LRD ∩ Reg.	Mech. Sparsity [24]	○	•		Cond. prior (passive) + L_1 mask sparsity (active)
	SlowVAE [67]	○	•		Temporal prior (passive) + Laplace sparse penalty
	CITRIS [68]	○	•		Intervention prior (passive) + factored assignment loss
	TDRL [12]	○	•		Shift prior (passive) + modular mechanism loss

Open question. ***What should factoredness mean for a billion-parameter text-to-image model***, and which theoretical guarantees survive the absence of canonical ground-truth factors?

H.2 MCL as Fine-tuning of Large Pretrained Models

Motivation. Pretraining of large generative models is necessarily ***passive***: the conditioning signal is whatever caption, label, or action happens to accompany each datapoint at web scale, and no explicit objective enforces mechanism contrast. Fine-tuning is where domain-specific expert knowledge enters the model, through methods such as LoRA [70], ControlNet [71], and DreamBooth [72]. MCL complements these adaptation methods with an objective that contrasts mechanism-gradient features across conditions.

Position. MCL offers an additional fine-tuning objective for pretrained conditional generators. This paper evaluates it in MDDiff and in matched Wan 1.4B world-model comparisons (Section 5). Extending these evaluations to broader tasks requires separating effects of the objective from training duration, architectural additions, and changes in the data distribution.

Open questions. Which layers of a U-Net or DiT should host the projection heads? How should MCL be balanced against the original denoising loss? Is parameter-efficient fine-tuning of the projection-head subspace alone sufficient?

H.3 Multi-objective, Multi-auxiliary Mechanism Contrast

Motivation. Realistic generative pipelines pair ***several auxiliary variables*** (text caption, robot action, audio cue, regime index) with ***several generative objectives*** (image, video, control trajectory, language). Examples include video-action world models [60], manipulation diffusion policies [63], and decision-as-generation pipelines [61, 62]. MCL as defined in Section 4 contrasts one mechanism gradient against one conditioning embedding, which is well-matched to single-auxiliary, single-objective settings but leaves the multi-axis setting open.

Open questions. Should each auxiliary $\mathbf{u}^{(k)}$ carry its own projection head and InfoNCE term, or should the contrastive signal be ***jointly normalized*** across auxiliaries? Does the sufficient-GMD condition decompose objective-by-objective, or does identifiability require simultaneous diversity across all $\mathbf{u}^{(k)}$? How does the mechanism-shift principle behave when auxiliaries are ***causally entangled***, e.g. an action partially explains the next observation while a text caption only describes high-level semantics?

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 distinguish the GMD identifiability result within its stated geometric model class (Theorem 1), the practical MCL objective (Section 4), and empirical evaluation on disentanglement benchmarks, Robotwin, and DROID (Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 3.2 specifies the geometric assumptions of Theorem 1: shared factor response directions, pairwise distinct mechanism profiles, an injective reference tangent-space map, and noiseless proper embeddings. Appendix C.6 analyzes the relationship between critic features and generator geometry. The empirical study covers three 64×64 disentanglement benchmarks and prediction with a Wan 1.4B backbone on Robotwin and DROID; world-model prediction metrics do not measure factor identifiability. Section 6 and Appendix H discuss further scope limits and open directions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 states the model class and geometric assumptions required for a shared, componentwise nonlinear correspondence. Formal statements and proofs are provided in Appendices B and C. The capacity comparison is restricted to its specified model classes, and the temporal specialization retains the main theorem's assumptions. Theorem 2 analyzes projected-feature diversity with fixed uniform keys.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Appendix D describes the synthetic and image-model training protocols and the world-model architecture, while Appendix E provides reference code for the MCL loss and helper modules. The world-model experiments are not fully specified: the paper does not give all training hyperparameters, update counts, sampling settings, and execution details needed to reproduce the Robotwin and DROID results. The disclosed information therefore supports partial rather than complete experimental reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The submission provides a project-code link and reference MCL implementation in Appendix E, but does not document a complete set of training and evaluation scripts, environments, and execution instructions for all reported results. The existing datasets used are `Shapes3D`, `MPI3D`, `Cars3D`, Robotwin, and DROID, cited at their usage points.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Section 5 specifies the datasets, comparisons, and metrics. Appendix D reports the synthetic and MDDiff training settings, including the two-stage image-model schedule. However, it does not report the complete optimizer, learning-rate, batch-size, update-count, loss-weight, and sampling configuration for the world-model experiments. The DROID comparison specifies which settings are held fixed but not all of their values.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports mean $\pm$ standard deviation over 15 runs, and Figure 5 shows synthetic MCC distributions over 30 random seeds per configuration. These summaries describe variation across the reported runs, not confidence intervals. Robotwin and DROID results are point estimates; training-seed variability and significance tests are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix D reports one NVIDIA H100 GPU for MDDiff and approximate Stage-2 fine-tuning times of 3 h on `Cars3D`, 4 h on `Shapes3D`, and 7 h on `MPI3D`. It also gives the image-model batch size and training schedule. Hardware memory, world-model training resources and runtimes, and total compute across all seeds, baselines, and exploratory runs are not fully reported. The per-dataset fine-tuning times are not estimates of total project compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The study uses existing public benchmarks: `Shapes3D`, `MPI3D`, `Cars3D`, Robotwin, and DROID. We do not recruit participants or collect a new dataset.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work studies identifiability for conditional generative models. Better factor recovery can support more interpretable and controllable generation. Improvements in controllable generation can also be misused to produce deceptive synthetic media. In robotics, good prediction scores do not establish safe closed-loop behavior, and further evaluation would be needed before deployment. The reported experiments do not establish robustness or safety outside their evaluation settings.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work introduces a training objective and a reference implementation, rather than a new scraped dataset or a general-purpose pretrained model release. Experiments use existing public benchmarks and pretrained backbones.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The datasets, baseline methods, and Wan backbone are credited through their original references, including Robotwin [51] and DROID [52]. The paper does not provide a complete inventory of the versions, licenses, and terms of use for every code, dataset, and model asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new methods (MD-VAE, MDDiff, ACG-WM, and the MCL objective) are documented in Section 4 and Appendix D, with a self-contained reference PyTorch implementation of MCL in Appendix E (Algorithms 1 and 2).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We use existing public datasets and do not conduct a new crowdsourcing study or recruit participants for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The reported experiments reuse existing datasets and do not involve new participant recruitment or direct interaction with human research subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLM assistance supported exploration and revision of theoretical assumptions, construction and checking of proof arguments, and manuscript editing. The reported training and evaluation pipelines contain no LLM component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.