VOCABSCOPE: READING CONCEPTS FROM VIDEO GENERATION STATES IN SPACE AND TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generation models can produce rich and dynamic videos from natural-language prompts, yet their hidden states remain difficult to interpret in terms of concepts. We introduce VocabScope, a framework that maps hidden states of video generators into vocabulary space while preserving their correspondence to video locations. Our method first establishes token-level correspondence between the video generator and a frozen vision–language model, then aligns their representation spaces, and finally connects intermediate representations in its language backbone to vocabulary predictions through layer-specific Jacobian transport. This makes it possible to examine what concept evidence is readable from a generation state and associate it with locations in video space and time. VocabScope offers a direct vocabulary-space view of video-generation hidden states and provides a tool for inspecting concept evidence at corresponding video locations.

1 INTRODUCTION

Recent video generation models, such as Sora (Brooks et al., 2024) and Wan (Wan Team, 2025), can synthesize rich and dynamic videos from natural-language prompts. However, a prompt only describes the intended content and does not reveal what a particular generation process has formed internally at a given generation step. Even with the same prompt, different generations can evolve toward different visual states as the video is progressively formed. Knowing what the prompt asks for is therefore not the same as knowing what a particular hidden state represents.

The final video likewise shows only the generation outcome, but does not reveal the concept evidence in intermediate hidden states or its correspondence to video locations. Even when intermediate video latents can be decoded, the resulting images are produced through the model's visual output pathway and do not directly reveal which concepts are represented in the hidden states themselves. This leads to a simple question: can we read a video-generation state in terms of human-interpretable concepts, while retaining their correspondence to locations in video space and time? In other words, we want to know what a video generator's hidden state says and where it says it.

Prior work has localized predefined concepts in diffusion and video models (Helbling et al., 2025; Jun et al., 2026) and discovered interpretable latent features (Kim et al., 2025b; Tinaz et al., 2025). However, these approaches do not express video-generator hidden states in a pretrained language-model vocabulary while preserving their correspondence to locations in space and time. Achieving this is challenging because video generators do not have a native vocabulary output. One option is to train a dedicated concept decoder on generator hidden states. Another is to decode intermediate video latents into images and analyze the resulting images with a vision–language model such as CLIP (Radford et al., 2021). The first route requires specifying the concepts to predict and the training supervision, while the second is limited by noisy intermediate reconstructions, which make concepts difficult to identify. Our goal is to read concept evidence directly from generator hidden states using the vocabulary space of a pretrained vision–language model. This requires mapping generator features into the model's visual representation space while retaining their correspondence to video locations. Preserving this correspondence allows us to examine both what concepts are readable and where their evidence is expressed across video space and time.

To this end, we propose VocabScope. To retain the video locations associated with the predicted concepts, we establish spatiotemporal correspondence between the generator token grid and the native video-token grid of a frozen vision–language model. We bridge the gap between their feature

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

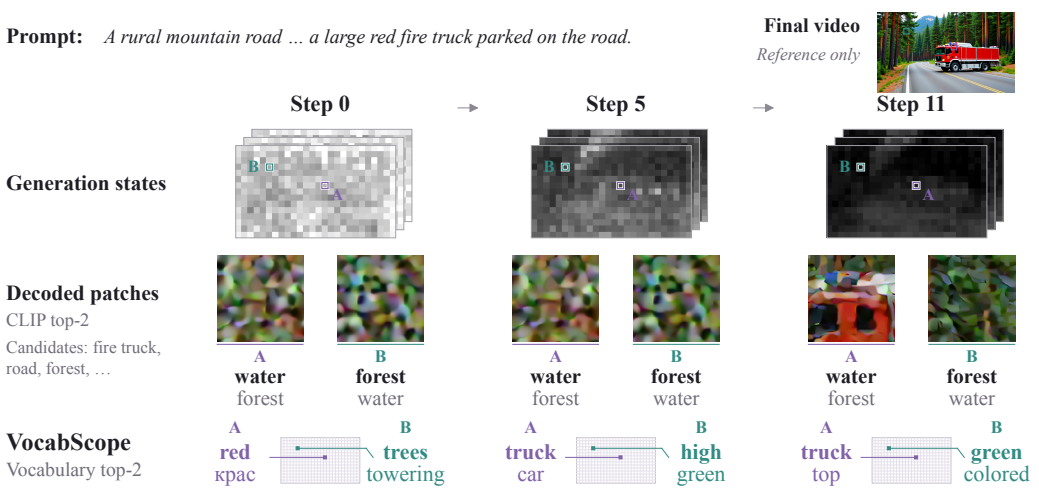


Figure 1: **Reading concepts directly from video-generation states.** Noisy decoded patches can obscure concepts that are readable from hidden states. At the truck-associated location, CLIP predicts *water* and *forest* from the decoded patches, whereas VocabScope reads *red* at the early step and *truck* at later steps. The first row visualizes hidden states using a shared PCA projection. The following rows compare CLIP predictions and VocabScope vocabulary predictions at the same two video locations.

spaces with a feature alignment layer trained through paired visual-feature alignment and output-distribution distillation. Reading intermediate representations in its language backbone through the frozen vocabulary head requires alignment with the final representation space, which we achieve through layer-specific average Jacobian transport (Gurnee et al., 2026). Together, these components produce vocabulary scores indexed by video position that allow us to discover concept evidence in generator states and associate it with video locations.

Our main contributions are summarized as follows:

- We introduce VocabScope, which exposes video-generation hidden states in vocabulary space while preserving their correspondence to video locations.
- We establish spatiotemporal correspondence between the video generator and the vision–language model, align their representation spaces, and connect the aligned features to vocabulary-level predictions through Jacobian transport.
- We evaluate concept predictions from intermediate generator states and their correspondence to video regions, and show that multi-position training improves average concept support across the evaluated blocks and steps.

2 Related Work

Concept representations in generative models. Internal features of generative models contain information useful beyond image synthesis. Probes recover depth and foreground structure from these features (Chen et al., 2023), while analyses of denoising trajectories reveal a progression from global composition to local appearance (Yi et al., 2024). Sparse autoencoders identify interpretable features and track their evolution across steps, layers, and architectures (Tinaz et al., 2025; Surkov et al., 2025; Kim et al., 2025b). Diffusion features also support correspondence between image regions and learned control (Tang et al., 2023a; Luo et al., 2023; 2024). In video generators, internal representations encode motion and temporal correspondence useful for control, recognition, and tracking (Xiao et al., 2024; Vélez et al., 2025; Nam et al., 2025). Recent work further examines attention specialization and reasoning dynamics across denoising steps and model layers (Liu et al., 2025; Wang et al., 2026). Studies of Stable Diffusion also show that concept information recovered by probes can be dominated by CLIP conditioning (Braunstein et al., 2025). Together, these studies motivate inspecting generator states while testing whether the recovered information depends on the particular state and its spatial arrangement.

Concept localization and intervention. Cross-attention and information-theoretic methods associate generated regions with conditioning tokens (Tang et al., 2023b; Kong et al., 2024). ConceptAttention and Seg4Diff construct open-vocabulary spatial maps from diffusion-transformer attention features (Helbling et al., 2025; Kim et al., 2025a). ConceptAttention also extends to video generation, while IMAP (Jun et al., 2026) produces spatiotemporal maps for object and motion concepts. Causal interventions identify layers or features associated with specified concepts and test their effects on generated content (Basu et al., 2024; Shi et al., 2025). These methods primarily ask where a supplied concept is represented or where to intervene to change it. VocabScope addresses the complementary question of which vocabulary tokens are supported at a given video location. This makes concept prediction at each position the starting point for examining the state.

Language-based interpretation of hidden states. Language-based methods make hidden representations interpretable through vocabulary scores or generated descriptions. Layerwise lenses map language-model activations to vocabulary predictions (Belrose et al., 2023; Pal et al., 2023). Patchscopes and LatentQA produce natural-language explanations of hidden representations (Ghandeharioun et al., 2024; Pan et al., 2024), while Activation Oracles train language models to answer questions about another language model's activations (Karvonen et al., 2025). Diffusion Lens interprets text-encoder states in text-to-image systems (Toker et al., 2024), and studies of vision–language models examine how visual features become aligned with vocabulary coordinates (Neo et al., 2025). J-Lens uses corpus-averaged Jacobians to estimate how intermediate activations can affect present or future vocabulary predictions (Gurnee et al., 2026). We build on this vocabulary-projection approach to interpret video-generation hidden states, preserving their video positions when connecting generator features to the frozen vision–language model.

3 METHOD

VocabScope expresses concept evidence in video-generation hidden states as vocabulary scores at corresponding video locations. Section 3.1 formulates this task and defines the generator feature grid. Section 3.2 establishes temporal and spatial correspondence between the generator and the vision–language model. Section 3.3 aligns their feature spaces through visual-feature alignment and output-distribution distillation. Section 3.4 maps intermediate language-backbone representations into the final representation space for vocabulary prediction.

3.1 PROBLEM FORMULATION

At denoising step s and generator block b, the hidden state is represented by a grid of feature vectors

$$\mathbf{H}_{s,b} = [\mathbf{h}_{s,b,t,y,x}]_{t,y,x} \,. \tag{1}$$

The grid has size $T \times H \times W$, where T is the number of temporal positions in the compressed video representation, and H and W are the numbers of spatial token rows and columns. Each entry $\mathbf{h}_{s,b,t,y,x}$ is a feature vector at temporal position t in the compressed video representation, row y, and column x. At each step s, the grid spans the entire video, so s indexes denoising progress while t indexes temporal positions in the compressed video representation.

The frozen pretrained vision–language model, denoted by M, has vocabulary $\mathcal{V}$. VocabScope assigns a score $a_{s,b,\ell,t,y,x}(v)$ to every vocabulary token $v \in \mathcal{V}$ at each aligned video position (t, y, x), where ℓ indexes a layer of the language backbone. Video tokens are feature vectors assigned to video locations, whereas vocabulary tokens are entries in $\mathcal{V}$.

3.2 SPATIOTEMPORAL CORRESPONDENCE

To associate each concept with a video location, generator features must occupy corresponding temporal and spatial positions in the vision–language model. The temporal length of the video-token grid of the vision–language model is first matched to that of the generator grid. The generator state contains T temporal positions in the compressed video representation. To obtain the same number of temporal positions in the vision–language model, the number of sampled video frames is adjusted according to its temporal compression ratio. The resulting T positions are paired with the temporal positions in the compressed video representation in temporal order.

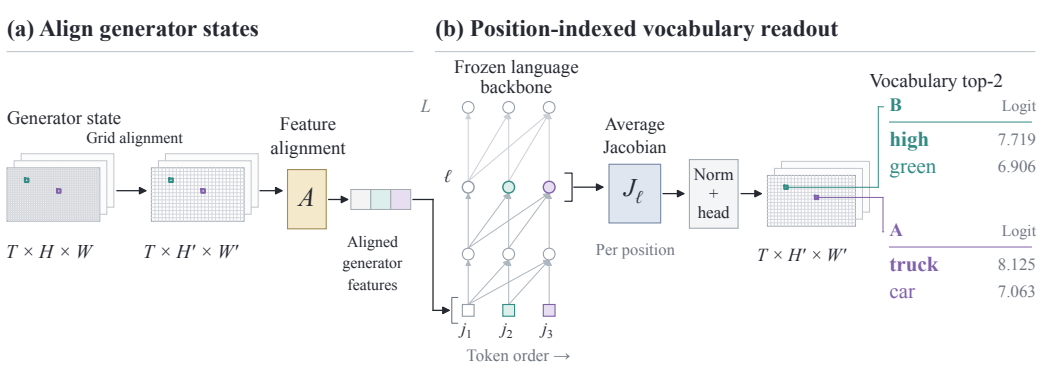


Figure 2: **VocabScope overview.** (a) Grid and feature alignment map generator states into the visual input space of a frozen vision–language model while preserving video locations. (b) Jacobian transport connects intermediate representations to the frozen vocabulary head through final normalization, producing vocabulary scores at each position. Colors indicate corresponding video locations, with example top-2 predictions shown on the right.

Let $H' \times W'$ denote the spatial grid size of the video tokens in the vision–language model, with H' rows and W' columns. At each matched temporal position t, generator features are mapped from the $H \times W$ grid onto this $H' \times W'$ grid. For each target position (y, x), let $\mathcal{R}_{y,x}$ be the nonempty set of generator positions in the corresponding spatial region. The feature at this target position is obtained by averaging these generator features

$$\widetilde{\mathbf{h}}_{s,b,t,y,x} = \frac{1}{|\mathcal{R}_{y,x}|} \sum_{(i,j)\in\mathcal{R}_{y,x}} \mathbf{h}_{s,b,t,i,j}. \tag{2}$$

Here $|\mathcal{R}_{y,x}|$ is the number of generator positions in the region. The average combines features within each assigned spatial region without mixing features from different temporal positions in the compressed video representation. The resulting grid $\widetilde{\mathbf{H}}_{s,b}$ has size $T \times H' \times W'$. Subsequent coordinates (t, y, x) refer to this aligned grid.

3.3 Cross-model Representation Alignment

After grid alignment, the features remain in the generator representation space. A shared affine feature alignment layer, denoted by A, maps them into the visual input space of the vision–language model

$$\mathbf{z}_{s,b,t,y,x} = A(\widetilde{\mathbf{h}}_{s,b,t,y,x}) = \mathbf{W}_A \widetilde{\mathbf{h}}_{s,b,t,y,x} + \mathbf{a}_A. \tag{3}$$

Here $\mathbf{W}_A$ and $\mathbf{a}_A$ are the trainable weight matrix and bias. The generator features $\widetilde{\mathbf{h}}_{s,b,t,y,x}$ are mapped to $\mathbf{z}_{s,b,t,y,x}$, which replace the native visual features of the vision–language model at the corresponding video-token positions in the language-backbone input. During this replacement, the text embeddings, sequence order, and positional indices remain unchanged.

Training uses generator hidden states from multiple denoising steps and generator blocks, with one feature alignment layer shared across these positions. The completed video is passed through the frozen vision–language model to extract native visual features as alignment targets. For each training video i, the objectives below use features from a selected step and block. We omit s and b for notational simplicity.

For training video i, each native visual feature $\mathbf{z}^{i,\mathrm{nat}}_{t,y,x}$ is paired with the generator feature $\widetilde{\mathbf{h}}^{i}_{t,y,x}$ at the corresponding position on the aligned grid. The initial training objective aligns the transformed generator features with these native visual features using cosine distance. For N training videos with the same aligned grid size, the loss is

$$\mathcal{L}_{\mathrm{emb}} = \frac{1}{NTH'W'} \sum_{i=1}^{N} \sum_{t,y,x} \left[1 - \cos\left(A(\widetilde{\mathbf{h}}^{i}_{t,y,x}), \mathbf{z}^{i,\mathrm{nat}}_{t,y,x}\right)\right]. \tag{4}$$

The cosine term compares the directions of the two feature vectors, and the loss decreases as they align.

The cosine loss in Equation 4 aligns feature directions but does not ensure consistent language predictions. To address this limitation, the feature alignment layer is refined through output-distribution distillation (Hinton et al., 2015). The frozen vision–language model first generates a description using native visual features from the completed video. At each description token, we match the next-token distributions obtained with native visual features and aligned generator features, conditioned on the same preceding description tokens.

For training video i at caption position m, these next-token distributions over the full vocabulary are denoted by $p_{i,m}$ and $q_{i,m}$, respectively. Refinement minimizes their KL divergence together with the feature-alignment loss

$$\begin{aligned} \mathcal{L}_A &= \mathcal{L}_{\text{emb}} + \lambda_{\text{KD}} \mathcal{L}_{\text{KD}}, \\ \mathcal{L}_{\text{KD}} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\tau^2}{K_i} \sum_{m=1}^{K_i} D_{\text{KL}}(p_{i,m} \| q_{i,m}) \,, \end{aligned} \tag{5}$$

Here K_i is the number of generated description tokens used for distillation, τ is the shared softmax temperature, and λ_{KD} controls the distillation weight. Both predictions use the same description instruction and preceding caption tokens, so only the visual features differ.

3.4 Position-indexed Vocabulary Readout

The aligned generator features produce intermediate representations in the language backbone, whereas the vocabulary head is trained on final-layer representations. To read these intermediate representations in vocabulary space, VocabScope follows J-Lens (Gurnee et al., 2026) and uses a layer-specific Jacobian transport matrix to map them into the final representation space.

For a generator state at step s and block b, let $\mathbf{r}_{\ell,j}$ denote the residual vector at layer ℓ and video-token position $j = j(t, y, x)$, corresponding to video location (t, y, x). Let L denote the final layer before output normalization. The first-order dependence of the final-layer representation $\mathbf{r}_{L,j'}$ at position j' on the intermediate representation $\mathbf{r}_{\ell,j}$ is described by the Jacobian

$$\frac{\partial \mathbf{r}_{L,j'}}{\partial \mathbf{r}_{\ell,j}}. \tag{6}$$

This matrix describes how changes in intermediate feature directions propagate to the final representation space, with other layer-ℓ residuals held fixed.

Information from one intermediate position can propagate to multiple final-layer positions through self-attention. The Jacobians are therefore summed over the selected final-layer positions to aggregate the influence of that intermediate position on these outputs. Averaging these aggregated Jacobians over visual source positions and calibration inputs gives a shared transport matrix J_ℓ for layer ℓ. For calibration input i, write the residual as $\mathbf{r}^i_{\ell,j}$. Let $\mathcal{I}^i_{\text{src}}$ and $\mathcal{I}^i_{\text{tgt}}$ contain the visual source positions and selected final-layer target positions, respectively. For N_J calibration inputs, the transport matrix is

$$J_\ell = \frac{1}{N_J} \sum_{i=1}^{N_J} \frac{1}{|\mathcal{I}^i_{\text{src}}|} \sum_{j \in \mathcal{I}^i_{\text{src}}} \sum_{j' \in \mathcal{I}^i_{\text{tgt}}} \frac{\partial \mathbf{r}^i_{L,j'}}{\partial \mathbf{r}^i_{\ell,j}}. \tag{7}$$

Each J_ℓ is estimated once and reused across video locations and generation states. The layer-specific Jacobian transport matrix maps the residual $\mathbf{r}_{\ell,j}$ at each video location to $J_\ell \mathbf{r}_{\ell,j}$ in the final representation space of the language backbone. The frozen final normalization and vocabulary head then give

$$a_{s,b,\ell,t,y,x}(v) = \mathbf{u}_v^\top \, \text{Norm}_M \big(J_\ell \mathbf{r}_{\ell,j(t,y,x)} \big) \,, \qquad v \in \mathcal{V}, \tag{8}$$

where Norm_M is the frozen final normalization and $\mathbf{u}_v$ is the vocabulary-head vector for token v. The scores $a_{s,b,\ell,t,y,x}(v)$ express concept evidence in a generator hidden state in vocabulary space while retaining the corresponding video locations.

4 Experiments

We evaluate whether VocabScope predicts recognizable concepts from intermediate video-generation states and whether these predictions match the corresponding video regions. We compare against feature controls, CLIP-based baselines, and IMAP, then examine the effects of training coverage, training objectives, and Jacobian transport through ablation studies.

4.1 Experimental Setting

Models and data. We implement VocabScope with AnyFlow–Wan2.1-T2V-1.3B (Gu et al., 2026; Wan Team, 2025) as the video generator and Qwen3-VL-2B-Instruct (Bai et al., 2025) as the frozen vision–language model. Both models remain frozen. Our dataset contains 280 generated videos, partitioned by prompt family into 196 training videos, 42 validation videos, and 42 test videos. Each video contains 81 frames at a resolution of 480×832 and is generated with four denoising steps.

Implementation details. We train one feature alignment layer shared across generator blocks 5, 10, 20, and 28 and denoising steps 0, 1, 2, 3, and keep it fixed during evaluation. From the $21 \times 30 \times 52$ generator grid, we retain temporal positions 2, 10, and 18 and apply 2×2 spatial average pooling to obtain a $3 \times 15 \times 26$ grid. The alignment layer maps the 1,536-dimensional generator features to the 2,048-dimensional visual feature space of the vision–language model. For vocabulary prediction, the vision–language model receives the fixed instruction "Describe this video." without the original generation prompt or reference caption.

Training details. The feature alignment layer is trained for 50 epochs with the cosine feature-alignment loss, followed by 20 epochs with the combined feature-alignment and output-distillation loss. Each epoch visits every training video once and selects one block–step combination per video, cycling through the training combinations across epochs. The distillation weight is $\lambda_{\mathrm{KD}} = 0.1$, and the temperature is $\tau = 1$. Both stages use AdamW (Loshchilov & Hutter, 2019) with a constant learning rate of 10^{-4}.

Jacobian estimation. With the trained alignment layer frozen, we estimate one shared average Jacobian transport matrix from the aligned visual embeddings at the language-backbone input to the final residual representation at layer 27. Each of the 196 training videos contributes one predefined block–step combination, distributed approximately evenly across blocks 5, 10, 20, and 28 and denoising steps 0, 1, 2, 3. Reference captions are excluded from the input. The matrix remains fixed across the evaluated generator blocks, denoising steps, and held-out videos.

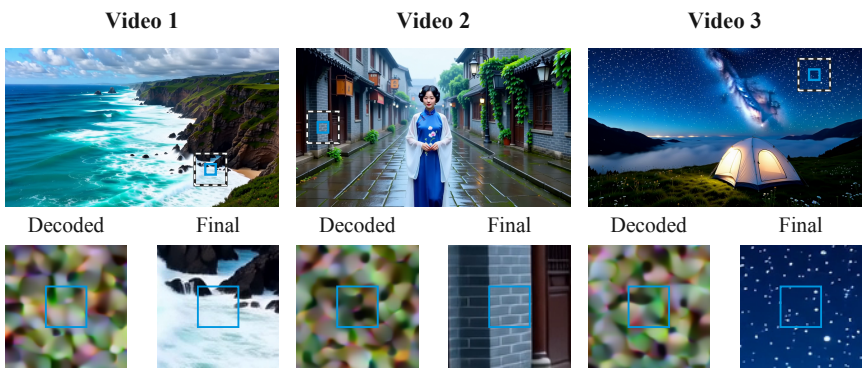
Evaluation metrics. For each evaluated position, we inspect the five highest-ranked words and judge whether each word is supported by the corresponding region and its surrounding context in the completed video, supported only elsewhere in the video, unsupported, or uncertain. **Local support@5** is the percentage of these five words supported by the corresponding region. For example, three locally supported words out of five give a local support@5 of 60%. **Video support@5** also counts words supported elsewhere in the video. We average these percentages over the same evaluated positions for every method. Unsupported and uncertain words remain in the denominator. These metrics measure agreement with the completed video, which serves as a visual reference rather than ground truth for which concepts are represented in an intermediate state.

Baselines. We compare VocabScope with two CLIP-based methods (Radford et al., 2021) and an IMAP-based method (Jun et al., 2026). **Decoded patch + CLIP** ranks concepts from a patch decoded from the intermediate video latent at the evaluated step. **Final patch + CLIP** uses the corresponding patch from the completed video as a reference for recognition after generation. **IMAP (GramCol)** is adapted to rank candidate concepts using spatial maps computed from generator attention features. All methods rank the same fixed candidate vocabulary constructed from training prompts. Appendix A provides the baseline details.

4.2 Spatial Correspondence of Vocabulary Predictions

We examine whether vocabulary predictions match their associated video regions when intermediate reconstructions remain noisy. We use matched positions in held-out videos at denoising step 1, with VocabScope evaluated at generator block 10. Existing annotations are retained after manual review and correction.

Dependence on Generator States. We compare the original generator features with three controls. Spatial shuffling within each temporal position tests dependence on spatial arrangement. Features from another video test dependence on the source state, and zero features test predictions without



	Video 1	Video 2	Video 3
Decoded patch + CLIP	particle, particles, blurred, pattern, dimensional	blurred, particles, pattern, particle, beads	particles, particle, beads, fabric, blurred
Final patch + CLIP	waves, wave, ocean, surf, sea	doors, door, sidewalks, walls, windows	stars, snow, frozen, sky, clusters
IMAP (GramCol)	ocean, shoreline, beach, seaside, waterfall	street, castle, sidewalk, rustic, cottage	sunrise, sunlight, mountain, valley, waterfall
VocabScope	**crashing, breaking, waves, washing, wash**	**brick, grey, gray, walls, buildings**	**stars, bright, sparkling, clouds, countless**

Figure 3: **Qualitative comparison at matched video positions.** Each column shows a final-frame reference, decoded and final patches, and ranked top-5 predictions. Blue boxes mark evaluated positions and dashed boxes mark CLIP crops.

generator-state information. Only the input features change, with the prediction model and reference regions held fixed. Table 1 reports results over the full vocabulary.

Table 1: Support of top-5 predictions from the full vocabulary under generator-feature controls.

Generator features	Local support@5 (%)	Video support@5 (%)
Generator features	**55.4**	**57.7**
Spatially shuffled features	23.1	50.8
Features from another video	20.0	34.6
Zero generator features	1.5	1.5

Generator features achieve 55.4% local support, showing that the hidden states contain readable concept information about the corresponding video regions. After spatial shuffling, local support drops to 23.1%, while video support remains at 50.8%. Many predicted words still describe the video, but fewer match the evaluated region. One possible explanation is that shuffling preserves the input feature content, which the vision–language model can still aggregate across positions through self-attention. Replacing the source features with those from another video reduces support, while zero features yield almost no supported predictions.

Comparison with baselines. Table 2 compares VocabScope with the CLIP and IMAP baselines over shared candidates. Missing labels are supplemented with AI-assisted judgments pending human verification.

Table 2: Top-5 concept support over shared candidates.

Method	Local support@5 (%)	Video support@5 (%)
Decoded patch + CLIP	0.0	0.0
Final patch + CLIP	63.8	69.2
IMAP (GramCol)	23.1	36.9
VocabScope	**68.5**	**69.2**

VocabScope achieves 68.5% local support, compared with no supported predictions from Decoded patch + CLIP at the evaluated step and 63.8% from Final patch + CLIP. VocabScope and Final patch + CLIP have the same video support.

Qualitative comparison. Figure 3 shows predictions from all four methods at the same video positions. VocabScope predicts waves for the coastal region, brick for the wall, and stars for the night sky, although the decoded patches remain highly noisy.

Concept predictions across denoising steps. Table 3 reports configuration C on the test dataset, using shared candidates and averaging over generator blocks 5, 10, 20, and 28. At step 0, local support is already 39.0%, indicating that predictions can match the corresponding regions in the completed video from the first generation step. Support varies across steps rather than increasing monotonically.

Table 3: Concept support across denoising steps.

Denoising step	0	1	2	3
Local support@5 (%)	39.0	47.9	43.8	41.2
Video support@5 (%)	48.5	54.2	50.0	49.6

4.3 Ablation Studies

We examine how training coverage, training objectives, and Jacobian transport affect concept predictions. All ablations use the same test dataset and candidate vocabulary, with local and video support averaged across generator blocks 5, 10, 20, and 28 and denoising steps 0, 1, 2, 3.

Training coverage. We compare three training configurations to examine how training coverage affects vocabulary predictions. Configuration A uses generator block 10 at step 1. Configuration B keeps block 10 fixed and includes steps 0, 1, 2, 3. Configuration C, used in our main setup, includes blocks 5, 10, 20, and 28 at each of these four steps. Configurations B and C use the same training budget. Each configuration has one shared feature alignment layer and a corresponding fixed Jacobian transport matrix from layer 26 to layer 27 of the language backbone. Table 4 compares the three configurations on test videos using the same candidate vocabulary and annotation protocol.

Table 4: Effect of training coverage on top-5 support.

Training coverage	Local support@5 (%)	Video support@5 (%)
A: block 10 $\times$ step 1	27.8	32.5
B: block 10 $\times$ steps 0, 1, 2, 3	32.4	39.6
C: blocks 5/10/20/28 $\times$ steps 0, 1, 2, 3	**43.0**	**50.6**

Training across all four steps increases mean local support from 27.8% to 32.4%. Extending training to four blocks further increases it to 43.0%, with mean video support reaching 50.6%. These results show that fitting the shared mapping and its corresponding Jacobian across multiple generator positions improves vocabulary predictions across the evaluated blocks and steps on held-out videos.

Training objectives. We compare feature alignment alone, output-distribution distillation alone, and the full two-stage objective under configuration C. All three use the same random initialization, training positions, and total number of optimizer updates. For each trained feature alignment layer, we estimate a corresponding Jacobian transport matrix using the same training videos and predefined positions. Table 5 reports support on test videos using the shared candidate vocabulary. The loss and Jacobian ablations reuse existing annotations and supplement missing labels with AI-assisted judgments pending human verification.

Table 5: Effect of training objectives on top-5 support.

Training objective	Local support@5 (%)	Video support@5 (%)
Feature alignment only	38.6	47.1
Output-distribution distillation only	10.7	16.0
Full objective	**43.0**	**50.6**

Feature alignment alone gives substantially higher support than distillation alone. Adding distillation to feature alignment increases local support from 38.6% to 43.0% and video support from 47.1% to 50.6%. This comparison suggests that feature alignment provides the main basis for predicting spatially supported concepts, while output-distribution distillation provides an additional improvement.

Jacobian transport. We compare predictions from language-backbone layers 1, 9, 17, and 26 with and without their corresponding Jacobian transport, keeping the feature alignment layer and test positions fixed. Both branches use the same final normalization and vocabulary head.

Table 6: Effect of Jacobian transport across language-backbone layers.

	Local support@5 (%)		Video support@5 (%)	
Layer	Without J	With J	Without J	With J
1	5.9	**32.8**	10.7	**46.3**
9	11.8	**34.8**	17.5	**46.5**
17	18.4	**38.3**	24.2	**48.4**
26	42.3	**43.0**	50.4	**50.6**

Reference region	Layer	Without J	With J
	1	fields, corridors, twilight	trees, forests, forest
	9	pose, lean, aging	trees, forest, tree
	17	trees, forests, climbing	trees, forest, tree
	26	trees, tall, towering	trees, tree, tall
	1	fields, ink, lake	reflections, water, reflected
	9	mud, underwater, freshwater	ponds, aquatic, water
	17	palm, swirling, like	ponds, pools, fluid
	26	water, clear, reflecting	water, clear, reflecting

Figure 4: **Vocabulary predictions across language-backbone layers.** Two reference frames with marked regions and enlarged insets are paired with top-3 predictions at generator block 10 and step 1.

Table 6 shows that Jacobian transport improves concept support most at earlier layers, while the difference becomes small near the final layer. This supports its role in making intermediate representations interpretable through the frozen vocabulary head. Figure 4 shows the same pattern qualitatively. With Jacobian transport, earlier layers predict concepts such as trees and water for the corresponding regions, while predictions with and without transport are similar at layer 26.

5 Conclusion and Limitations

VocabScope maps video-generation hidden states to vocabulary predictions while preserving their spatiotemporal positions. Our experiments show that these predictions capture concept evidence in intermediate states and retain spatial correspondence. Broader training coverage improves support across generation positions, while Jacobian transport makes earlier language-model layers more interpretable through the vocabulary head.

Our evaluation uses one generator–vision–language model pair and single-token concepts. Extending to other models and phrases remains future work. Self-attention can combine information across positions.

AI USE STATEMENT

We used generative AI models to produce the video dataset, as described in the experimental setting. AI tools also assisted with experiment design, method implementation, data cleaning, result analysis, figure preparation, and manuscript drafting and editing. For concept-support evaluation, AI-assisted judgments supplemented human annotations, with their review status reported in the experiments. The authors take responsibility for the final content, results, and claims of this work.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Samyadeep Basu, Keivan Rezaei, Priyatham Kattakinda, Vlad I. Morariu, Nanxuan Zhao, Ryan A. Rossi, Varun Manjunatha, and Soheil Feizi. On mechanistic knowledge localization in text-to-image generative models. In *International Conference on Machine Learning*, 2024.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Cameron Braunstein, Mariya Toneva, and Eddy Ilg. CLIP is all you need for human-like semantic representations in stable diffusion. *arXiv preprint arXiv:2511.08075*, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL https://openai.com/index/video-generation-models-as-world-simulators/.

Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv:2306.05720*, 2023.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning*, 2024.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Alec Helbling, Tuna Han Salih Meral, Benjamin Hoover, Pinar Yanardag, and Duen Horng Chau. ConceptAttention: Diffusion transformers learn highly interpretable features. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22946–22963, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Youngjun Jun, Seil Kang, Woojung Han, and Seong Jae Hwang. I'm a map! interpretable motion-attentive maps: Spatio-temporally localizing concepts in video diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525–11535, 2026.

Adam Karvonen, James Chua, Clément Dumas, Kit Fraser-Taliente, Subhash Kantamneni, Julian Minder, Euan Ong, Arnab Sen Sharma, Daniel Wen, Owain Evans, and Samuel Marks. Activation oracles: Training and evaluating LLMs as general-purpose activation explainers. *arXiv preprint arXiv:2512.15674*, 2025.

Chaehyun Kim, Heeseong Shin, Eunbeen Hong, Heeji Yoon, Anurag Arnab, Paul Hongsuck Seo, Sunghwan Hong, and Seungryong Kim. Seg4Diff: Unveiling open-vocabulary semantic segmentation in text-to-image diffusion transformers. In *Advances in Neural Information Processing Systems*, 2025a.

Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Revelio: Interpreting and leveraging semantic information in diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable diffusion via information decomposition. In *International Conference on Learning Representations*, 2024.

Bingyan Liu, Chengyu Wang, Tongtong Su, Huan Ten, Jun Huang, Kailing Guo, and Kui Jia. Understanding attention mechanism in video diffusion models. *arXiv preprint arXiv:2504.12027*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023.

Grace Luo, Trevor Darrell, Oliver Wang, Dan B. Goldman, and Aleksander Holynski. Readout guidance: Learning control from diffusion features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. In *Advances in Neural Information Processing Systems*, 2025.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *International Conference on Learning Representations*, 2025.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning*, 2023.

Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. *arXiv preprint arXiv:2412.08686*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. URL https://arxiv.org/abs/2103.00020.

Yingdong Shi, Changming Li, Yifan Wang, Yongxiang Zhao, Anqi Pang, Sibei Yang, Jingyi Yu, and Kan Ren. Dissecting and mitigating diffusion bias via mechanistic interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Advances in Neural Information Processing Systems*, 2023a.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023b.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Michael Toker, Hadas Orgad, Mor Ventura, Dana Arad, and Yonatan Belinkov. Diffusion lens: Interpreting text encoders in text-to-image pipelines. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Pedro Vélez, Luisa F. Polanía, Yi Yang, Chuhan Zhang, Rishabh Kabra, Anurag Arnab, and Mehdi S. M. Sajjadi. From image to video: An empirical study of diffusion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ruisi Wang, Zhongang Cai, Fanyi Pu, Junxiang Xu, Wanqi Yin, Maijunxian Wang, Ran Ji, Chenyang Gu, Bo Li, Ziqi Huang, Hokin Deng, Dahua Lin, Ziwei Liu, and Lei Yang. Demystifing video reasoning. *arXiv preprint arXiv:2603.16870*, 2026.

Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. In *Advances in Neural Information Processing Systems*, 2024.

Mingyang Yi, Aoxue Li, Yi Xin, and Zhenguo Li. Towards understanding the working mechanism of text-to-image diffusion model. In *Advances in Neural Information Processing Systems*, 2024.

APPENDIX

A BASELINE DETAILS

All baselines rank the same 2,245 candidate words from training prompts and return the five highest-scoring words at each evaluated position.

CLIP baselines. Frozen CLIP ViT-B/32 (Radford et al., 2021) ranks candidates by image–text cosine similarity using the template "a photo of {word}". Each input is a 96×96 crop centered on the evaluated 32×32 cell in frame 8. Decoded patch + CLIP uses the VAE reconstruction of the noisy latent immediately before the generator forward at step 1. Final patch + CLIP uses the corresponding crop from the completed video.

IMAP. We adapt the GramCol component of IMAP (Jun et al., 2026) for candidate ranking. The frozen generator text encoder encodes each word independently. We average its subtoken features, excluding special tokens, and apply the generator text projection and key projection. These additional keys are used only for analysis and do not change generation.

Within each attention head and frame, query–key matching selects a visual token for each candidate. Inner products between its cross-attention output and those of other visual tokens produce a spatial score map. We standardize scores across candidates within each head, sum over heads, and average generator blocks 5 through 19 at step 1. We then apply a softmax across candidates and average each 2×2 group of spatial tokens to match the evaluated 32×32 cell.

B CONCEPT-GUIDED INTERVENTION

We use vocabulary scores to select regions associated with a target concept, modify the generator features at those locations, and continue generation. The following example removes a fire truck from a generated video.

Selecting regions. For target words $\mathcal{T}$ and scene words $\mathcal{B}$, the concept score is

$$C_{s,b,\ell,p} = \frac{1}{|\mathcal{T}|} \sum_{w \in \mathcal{T}} a_{s,b,\ell,p}(w) - \frac{1}{|\mathcal{B}|} \sum_{w \in \mathcal{B}} a_{s,b,\ell,p}(w), \tag{B.1}$$

where $a_{s,b,\ell,p}(w)$ is the vocabulary score at video location p, step s, generator block b, and language-backbone layer ℓ. For each step and temporal position, we average normalized spatial ranks across blocks 6–17 and language-backbone layers 0–21. The highest-ranked 40% of locations form an intervention mask, expanded to the generator grid. Masks are computed from the unmodified source trajectory. In this example, $\mathcal{T} = \{\text{fire}, \text{truck}, \text{vehicle}\}$ and $\mathcal{B} = \{\text{road}, \text{mountain}, \text{forest}\}$. Figure B.1(a) displays only the highest-scoring 10% for clarity. The intervention uses the full 40% mask.

Changing the state. An auxiliary branch replaces cross-attention value vectors for a source phrase with those for a target phrase throughout the generator. The update is

$$\mathbf{h}^{+} = \mathbf{h} + \beta M \odot (\mathbf{h}^{\text{aux}} - \mathbf{h}), \tag{B.2}$$

where $\mathbf{h}$ and $\mathbf{h}^{\text{aux}}$ are block outputs from the same input with original and modified cross-attention. The binary mask M is broadcast across feature channels, and β controls the update strength. Here, we replace "and a large red fire truck parked on the road" with "and no truck anywhere on the road" and update block 29 at every step with $\beta = 1$. This example uses a separate earlier configuration fitted at block 10 and step 8, with 12 denoising steps.

Comparison conditions. **No intervention** continues generation without a feature update. **Concept-guided intervention** applies the update at the selected regions. **Shifted mask** shifts this mask horizontally, and **Random mask** selects the same number of locations at random. The three intervention conditions use the same update rule and strength. Figure B.1(b) shows the results.



Figure B.1: **Concept regions and intervention results.** (a) Red cells show the highest-scoring 10% of locations at step 6 on completed source frames. (b) The intervention results use 40% masks. Concept-guided intervention removes the truck, while the shifted and random masks retain it. The road and surrounding scenery also change.

C Reproducibility Statement

An anonymous code package and a runnable demo are provided as supplementary material.